Fast and Accurate Texture Recognition with Multilayer Convolution and Multifractal Analysis

Hicham Badri, Hussein Yahia, Khalid Daoudi

Fast and Accurate Texture Recognition with Multilayer Convolution and Multifractal Analysis

Hicham Badri, Hussein Yahia, and Khalid Daoudi

INRIA Bordeaux Sud-Ouest, 33405 Talence, France
{hicham.badri,hussein.yahia,khalid.daoudi}@inria.fr

Abstract. A fast and accurate texture recognition system is presented. The new approach consists in extracting locally and globally invariant representations. The locally invariant representation is built on a multi-resolution convolutional network with a local pooling operator to improve robustness to local orientation and scale changes. This representation is mapped into a globally invariant descriptor using multifractal analysis. We propose a new multifractal descriptor that captures rich texture information and is mathematically invariant to various complex transformations. In addition, two more techniques are presented to further improve the robustness of our system. The first technique consists in combining the generative PCA classifier with multiclass SVMs. The second technique consists of two simple strategies to boost classification results by synthetically augmenting the training set. Experiments show that the proposed solution outperforms existing methods on three challenging public benchmark datasets, while being computationally efficient.

1 Introduction

Texture classification is one of the most challenging computer vision and pattern recognition problems. A powerful texture descriptor should be invariant to scale, illumination, occlusions, perspective/affine transformations and even non-rigid surface deformations, while being computationally efficient. Modeling textures via statistics of spatial local textons is probably the most popular approach to build a texture classification system [1,2,3,4,5,6,7]. Based on this Bag-of-Words architecture, these methods try to design a robust local descriptor. Distributions over these textons are then compared using a proper distance and a nearest neighbor or kernel SVMs classifier [8]. Another alternative to regular histograms consists in using multifractal analysis [9,10,11,12,13]. The VG-fractal method [9] statistically represents the textures with the full PDF of the local fractal dimensions or lengths, while the methods in [10,11,12,13] make use of the box-counting method to estimate the multifractal spectrum. Multifractal-based descriptors are theoretically globally invariant to bi-Lipschitz transforms that include perspective transforms and texture deformations. A different approach recently presented in [14] consists in building a powerful local descriptor by cascading wavelet scattering transformations of image patches and using a generative PCA classifier [15]. Unfortunately, while these methods achieve high accuracy on some standard benchmark datasets, little attention is given to the computational efficiency, which is crucial in a real-world system.

We present in this paper a new texture classification system which is both accurate and computationally efficient. The motivation behind the proposed work comes from the success of multifractal analysis [10,9,11,12,13]. Given an input texture, the image is filtered with a small filter bank for various filter orientations. A pooling operator is then applied to improve robustness to local orientation change. This process is repeated for different resolutions for a richer representation. This first step generates various low-pass and high-pass responses that form a *locally* invariant representation. The mapping towards the final descriptor is done via multifractal analysis. It is well known that the *multifractal spectrum* encodes rich texture information. The methods in [10,11,12,13] use the box-counting method to estimate the multifractal spectrum. However, this method is unstable due the limited resolution of real-world images. We present a new multifractal descriptor that is more stable and improves invariance to bi-Lipschitz transformations. This improvement is validated by extensive experiments on public benchmark datasets. The second part of our work concerns training strategies to improve classification rates. We propose to combine the generative PCA classifier [14,15] with kernel SVMs [8] for classification. We also introduce two strategies called ”synthetic training” to artificially add more training data based on illumination and scale change. Results outperforming the state-of-the-art are obtained over challenging public datasets, with high computational efficiency.

The paper is organized as follows : section 2 describes the proposed descriptor, section 3 presents the proposed training strategies, section 4 presents classification results conducted on 3 public datasets as well as a comparison with **9** state-of-the-art methods.

2 Robust Invariant Texture Representation

The main goal of a texture recognition system is to build an *invariant* representation, a mapping which reduces the large intra-class variability. This is a very challenging problem because the invariance must include various complex transformations such as translation, rotation, occlusion, illumination change, non-rigid deformations, perspective view, among others. As a result, two similar textures with different transformation parameters must have similar descriptors. An example is given in Figure 1. Not only the system should be accurate, but it should be also computationally efficient. Otherwise, its use in a real-world system would be limited due to the long processing time to extract the descriptor. Our goal in this paper is to build both an *accurate* and *fast* texture recognition system. Our Matlab non-optimized implementation takes around 0.7 second to extract the descriptor on a medium size image (480×640) using a modern laptop. The processing time can be further decreased by reducing the resolution of the image without sacrificing much the accuracy. This is due to the strong robustness of our descriptor to scale changes via accurate multifractal statistics that encode rich multi-scale texture information. We explain in this section how we build the proposed descriptor, the motivation behind the approach and the connection with previous work.

2.1 Overview of the Proposed Approach

The proposed descriptor is based on two main steps :

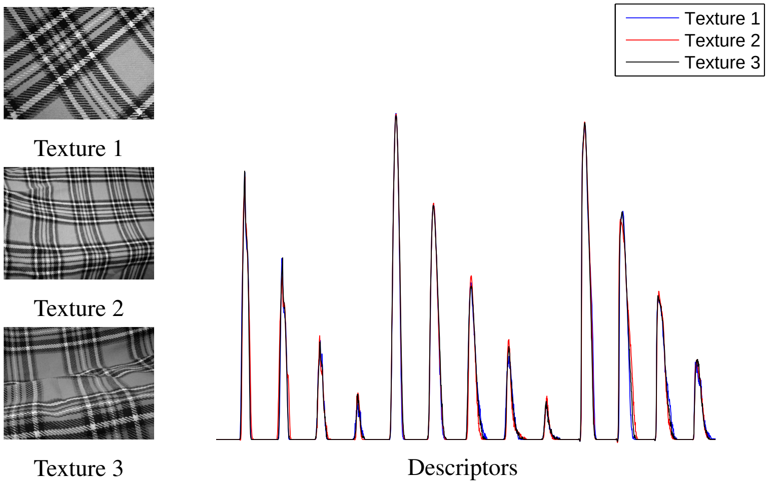

Fig. 1: Intra-class variability demonstration. The three textures 1, 2 and 3 exhibit strong changes in scale and orientation as well as non-rigid deformations. As can be seen, the proposed descriptor is nearly invariant to these transformations (see section 2).

1. Building a *locally* invariant representation : using multiple high-pass filters, we generate different sparse representations for different filter orientations. A pooling operator is applied on the orientation to increase the local invariance to orientation change. The process is repeated for multiple image resolutions for a richer representation.

2. Building a *globally* invariant representation : the first step generates various images that encode different texture information. We also include the multi-resolution versions of the input to provide low-pass information. We need a mapping that transforms this set of images into a stable, fixed-size descriptor. We use multifractal analysis to statistically describe each one of these images. We present a new method that extracts rich information directly from local singularity exponents. The local exponents encode rich multi-scale texture information. Their log-normalized distribution represents a stable mapping which is invariant to complex bi-Lipschitz transforms. As a result, the proposed multifractal descriptor is proven mathematically to be robust to strong environmental changes.

2.2 Locally Invariant Representation

A locally invariant representation aims at increasing the similarity of local statistics between textures of the same class. To build this representation, we construct a simple convolutional network where the input image is convolved with a filter bank for various orientations, and then pooled to reduce local orientation change. The multilayer extension consists in repeating the same process for various image resolutions on the low-pass output of the previous resolution, which offers a richer representation.

Given an input texture I, the image is first low-pass filtered with a filter ψ_l to reduce small image domain perturbations and produce an image $J_{1,0}$. This image is then filtered with multiple zero-mean high-pass filters $\psi_{k,\theta}$, where k denotes the filter number and θ its orientation. High-pass responses encode higher-order statistics that are not present in the low-pass response $J_{1,0}$. A more stable approach consists in applying the modulus on the high-pass responses, which imposes symmetric statistics and improves invariance of the local statistics. Applying multiple filtering with multiple different filters naturally increases the amount of texture information that are going to be extracted further via multifractal analysis. In order to increase the local invariance to orientation, we apply a pooling operator $\phi_\theta : \mathcal{R}^{i \times j \times n} \rightarrow \mathcal{R}^{i \times j}$ on the oriented outputs for each filter :

$$J_{1,k} = \phi_\theta(|J_{1,0} \star \psi_{k,\theta}| \, , \, \theta = \theta_1, ..., \theta_n) \ \ , \ \ k = 1, ..., K, \tag{1}$$

where n is the number of orientations and $i \times j$ is the size of the low-pass image. As a result, we obtain 1 low-pass response and K high-pass responses, each image is encoding different statistics. For a richer representation, we repeat the same operation for different resolutions $s = 2^{0,\dots,-L}$, where $s = 1$ is the finest resolution and $s = 2^{-L}$ is the coarsest resolution. The image generation process is then generalized as follows :

$$J_{s,k} = \begin{cases} I \star \psi_l & k = 0 \, , \, s = 1 \\ \downarrow (J_{2s,0} \star \psi_l) & k = 0 \, , \, s \neq 1 \\ \phi_\theta(|J_{s,0} \star \psi_{k,\theta}| \, , \, \theta = \theta_1, ..., \theta_n) & k = 1, ..., K, \end{cases} \tag{2}$$

where $\downarrow$ denotes the downsampling operator. We found that calculating statistics on multiple resolutions instead of a single one increases significantly the robustness of the descriptor. This can be expected because two textures may seem ”more similar” at a lower resolution. As a result, the intra-class variability decreases as the resolution decreases, but keeping higher resolution images is important to ensure extra-class decorrelation.

Dimensionality Reduction with Pooling

Using multiple filters $\psi_{k,\theta}$ increases dramatically the size of the image set. Knowing that each image $J_{s,k}$ will be used to extract statistics using multifractal analysis, this will result in a very large descriptor. One resulting issue is the high dimensionality of the training set. Another one is the processing time as the statistics should be applied on each image. We propose to merge different high-pass responses $J_{s,k}$ together to reduce the number of images. A straightforward approach would be to gather various images $\{J_{s,k} \, , \, k = t, .., u\}$ and then apply a pooling operator ϕ_r that is going to merge each image subset into one single image $J_{s,k_{t,..,u}}$:

$$J_{s,k_{t,..,u}} = \phi_r(\ J_{s,k} \, , \, k = t, .., u). \tag{3}$$

As a result, the number of high-pass responses will be decreased ; this leads to a reduced size descriptor. The pooling operator ϕ_r can be either the mean or the min/max functions. We take ϕ_r as a maximum function in this paper. An example is given in Figure 2 for one resolution $s = 0$ using 6 high-pass filters and one low-pass filter. The number of images is reduced from 7 to 3. For 5 resolutions ($s = 2^{0,\cdots,-4}$), the total number of images goes from 35 to 15, which is an important reduction.

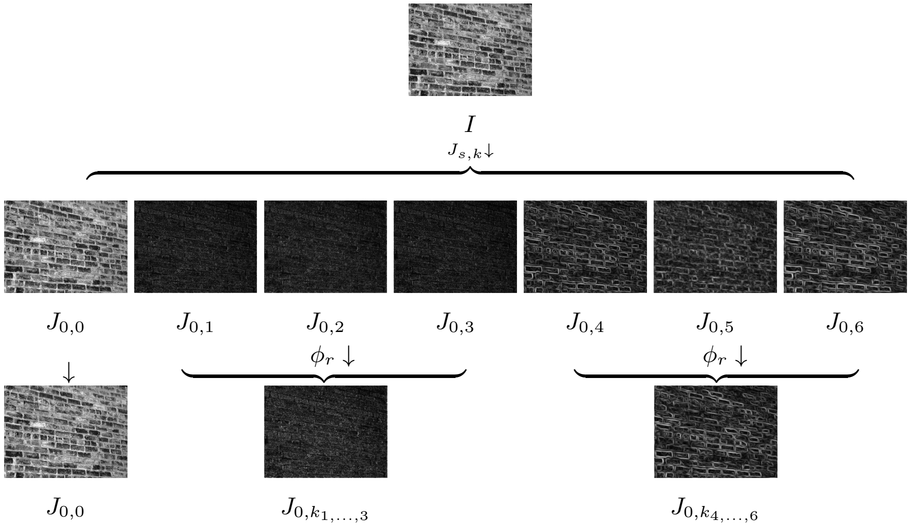

Fig. 2: Image generation example applied on the texture input I for one resolution using 6 high-pass filters. The images $J_{0,1\ldots6}$ are a result of the orientation pooling (eq. 2). The 6 images are reduced to 2 images using a pooling operator ϕ_r on similar responses to reduce the dimensionality. The same process is repeated for multiple resolutions.

2.3 Globally Invariant Representation

Once the set of low-pass and high-pass images is generated, we need to extract global statistics, a mapping into a fixed-size descriptor, which is *globally* invariant to the complex physical transformations. We propose to use a new multifractal approach to statistically describe textures suffering from strong environmental changes. To understand the difference between the proposed method and the previous work, we first present the standard fractal and multifractal analysis framework used by the previous methods, we then introduce the proposed approach.

Multifractal Analysis In a nutshell, a fractal object E is self-similar across scales. One characteristic of its irregularity is the so-called *box fractal dimension*. By measuring a fractal object on multiple scales r, the box fractal dimension is defined as a power-law relashionship between the scale r and the smallest number of sets of length r covering E [16]:

$$\dim(E) = \lim_{r\to 0} \frac{\log\ N(r,E)}{-\log\ r}, \tag{4}$$

Using squared boxes of size r, this dimension can be estimated numerically, known as the *box-counting* method. Multifractal analysis is an extension of this important notion. A multifractal object F is composed of many fractal components $F_{1,\ldots,f}$. In this case, a single fractal dimension is not sufficient to describe this object. The *multifractal spectrum* is the collection of all the associated fractal dimensions that describe the multifractal object.

It is easy to show mathematically that the fractal dimension is invariant to bi-Lipschitz transformations [17], which includes various transformations such as non-rigid transformations, view-point change, translation, rotation, etc.. As a result, the multifractal spectrum is also invariant to these transformations. This makes the multifractal spectrum an attractive tool to globally describe textures. However, the box-counting method gives a rather crude estimation of the real fractal dimension. The fractal dimension is estimated for each fractal set using a $\log$-$\log$ regression. As the resolution r is supposed to be very small ($r \to 0$), using small-sized boxes on a relatively low-resolution image results in a biased estimation due to the relatively low-resolution of real-world images [18]. It has been used as the core of various recent multifractal texture descriptors [10,11,12,13] that use the same box-counting method to build the final descriptor. We present a different method to statistically describe textures using multifractal analysis. Contrary to previous methods, we use a new measure which is based on the distribution of local singularity exponents. It can be shown in fact that this measure is related to the true multifractal spectrum, and its precision is proven by the high-accuracy of the proposed descriptor. Moreover, this approach is computationally efficient, which permits to achieve high accuracy at reduced processing time.

Proposed Multifractal Descriptor The proposed method first estimates the local singularity exponents $h(x)$ on each pixel x, and then applies the empirical histogram followed by $\log$ operator to extract the global statistics $\varphi_h = \log(\rho_h + \epsilon)$. This operation is performed on all the resulting images of the first step, which results in multiple histograms φ_{h_i}. The concatenation of all these histograms forms the final descriptor.

Let J be an image, and $\mu_\psi(B(x,r)) = \int_{B(x,r)} (J \star \psi_r)(y)dy$ a positive measure, where ψ_r is an appropriate wavelet at scale r (Gaussian in our case) and $B(x,r)$ a closed disc of radius $r > 0$ centered at x. Multifractal analysis states that the wavelet projections scale as power laws in r [19,20,21]. We use a microcanonical evaluation [20] which consists in assessing an exponent $h(x)$ for each pixel x :

$$\mu_\psi(B(x,r)) \approx \alpha(x) r^{h(x)} \ , r \to 0. \tag{5}$$

The validity of equation (5) has been tested on a large dataset [21], which proves that natural images exhibit a strong multifractal behavior. Introducing the $\log$, the formula is expressed as a linear fit :

$$\log(\mu_\psi(B(x,r))) \approx \log(\alpha(x)) + h(x)\log(r) \ , \ r \to 0. \tag{6}$$

Rewriting the equation in the matrix form permits to calculate all the exponents at once by solving the following linear system :

$$\underbrace{\begin{bmatrix} 1 & \log(r_1) \\ \vdots & \vdots \\ 1 & \log(r_l) \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \log(\alpha(x_1)) & \cdots & \log(\alpha(x_N)) \\ h(x_1) & \cdots & h(x_N) \end{bmatrix}}_{\eta} = \underbrace{\begin{bmatrix} \log(\mu_\psi(B(x_1,r_1))) & \ldots & \log(\mu_\psi(B(x_N,r_1))) \\ \vdots & \ldots & \vdots \\ \log(\mu_\psi(B(x_1,r_l))) & \ldots & \log(\mu_\psi(B(x_N,r_l))) \end{bmatrix}}_{b}, \tag{7}$$

$$\underset{\eta}{\operatorname{argmin}} \, ||A\eta - b||_2^2 \,, \; h(x_i) = \eta(2,i), \tag{8}$$

where N is the number of pixels of the image J, l is the number of scales used in the log-log regression. This matrix formulation is computationally efficient and plays an important role in the speed of the proposed method. Given the local exponents $h(x)$, which is an image of the same size of J that describes the local irregularities at each pixel, we need to extract now a fixed-size measure that globally describes the statistics of $h(x)$. Using the box-counting method, this would require extracting all the fractal fractal sets $F_h = \{x \,|\, h(x) \approx h\}$, and then calculating the box-counting dimension for each set F_h. As discussed before, this approach leads to a crude estimation of the true multifractal spectrum due to the actual low-resolution of real-world images. Moreover, a log-log regression should be performed on each fractal set. Instead, we propose to use the empirical histogram ρ_h followed by a log operator :

$$\varphi_h = \log(\rho_h + \epsilon), \tag{9}$$

where $\epsilon \geq 1$ is set to provide stability. The distribution of the local exponents is an invariant representation which encodes the multi-scale properties of the texture. The log acts as a normalization operator that nearly linearizes histogram scaling and makes the descriptor more robust to small perturbations. This way, we have access to reliable statistics [1]. This log-histogram is calculated on each image generated in the first step, which results in a set of histograms $\varphi_{h_{1,\ldots,M}}$, where M is the total number of generated images. The final descriptor φ is constructed by concatenating ($\biguplus$) all the generated histograms :

$$\varphi = \biguplus_{m}^{M} \varphi_{h_m}; \tag{10}$$

A descriptor example is given in Figure 3. This descriptor φ is the result of the concatenation of 14 log exponents histograms calculated on the images generated with the first step of the method presented in section 2.2 and further explained in Figure 2. Three images are generated for each scale s ; a low-pass response is presented in red, and two high-pass responses are presented in black and gray in the figure [2].

2.4 Analysis

The basic multifractal framework consists in generating multiple images and then extracting statistics using multifractal analysis. Multifractal descriptors are mathematically invariant to bi-Lipschitz transforms, which even includes non-rigid transformation and view-point change. The proposed method follows the same strategy, but is substantially different from the previous methods. The differences lie in both the image generation step and the statistical description. For instance, the WMFS method [13]

[1] A mathematical relationship between the log exponents histogram and the multifractal spectrum is presented in the supplementary material.

[2] A histogram was discarded for $s = 2^{-4}$ in the second high response (in gray) due to the large size of the filter which is larger than the actual size of the input image at resolution $s = 2^{-4}$.

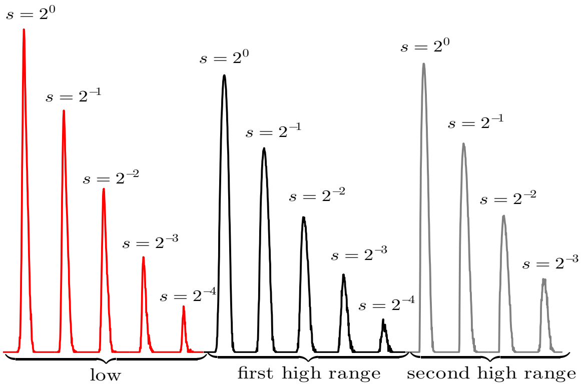

Fig. 3: A descriptor example using a low-pass response and two high-pass responses for 5 resolutions $s = 2^{0,\cdots,-4}$. The exponents log-histogram is calculated for each response and for multiple image resolutions s.

generates multiple images for multiple orientations, each oriented image is then analyzed using Daubechies discrete wavelet transform as well as using the wavelet leaders [22]. The multifractal spectrum (*MFS*) is then estimated for each image, for a given orientation using the box-counting method. Each *MFS* is then concatenated for a given orientation and the final descriptor is defined as the mean of all the descriptors over the orientation. Contrary to this method, we use different high-pass filters instead of one single analyzing wavelet, which permits to extract different statistics. Generating multiple descriptors for multiple orientations is computationally expensive. In contrast, we generate only one descriptor. To ensure local robustness to orientation, we apply a pooling operator on the *filtered responses*. This approach is much more computationally efficient. Finally, the core of our method is the new multifractal descriptor which permits to extract accurate statistics, contrary to the popular box-counting method as explained in the previous section. The proposed method takes about 0.7 second to extract the whole descriptor on an image of size 480×640, compared to 37 seconds as reported in the state-of-the-art multifractal method [13]. Experiments show that the proposed descriptor permits also to achieve higher accuracy, especially in large-scale situations when the extra-class decorrelation is a challenging issue.

2.5 Pre and Post Processing

Pre-processing and post-processing can improve the robustness of a texture recognition system. For instance, the method in [12] performs a scale normalization step on each input texture using blob detection. This step first estimates the scale of the texture and then a normalization is applied, which aims at increasing the robustness to scale change. Other texture classification methods such as [9] use Weber's law normalization to improve robustness to illumination. We do not use any scale normalization step such as [12,13], we rather use sometimes histogram equalization to improve robustness to illumination change. We also use a post-processing on features vector φ using wavelet domain soft-thresholding [**?**]. This step aims at increasing the intra-class correlation by reducing small histogram perturbations (for more details, please refer to the supplementary material).

3 Classification and Training Strategies

The second part of our work concerns the training aspect of the texture recognition problem. The globally invariant representation offers a theoretically stable invariant representation via accurate multifractal statistics. However, there are other small transformations and perturbations that may occur in real-world images and this is where a good training strategy will help us to take advantage of the proposed descriptor in practice. We work on two ideas :

1. The choice of the classifier can improve recognition rates : we introduce a simple combination between the Generative PCA classifier [14] and SVMs [8].
2. The lack of data is an issue, how to get more data? : Given an input training texture image, we synthetically generate more images by changing its illumination and scale. We call this strategy ”synthetic training”.

Experiments on challenging public benchmark datasets, including a large-scale dataset with 250 classes, validates the robustness of the proposed solution.

3.1 Classification

Support Vector Machines SVMs [8] are widely used in texture classification [10,12,12,13,7,6]. Commonly used kernels are mainly RBF Gaussian kernel, polynomials and χ^2 kernel. Extension to multiclass can be done via strategies such as one-vs-one and one-vs-all. In this paper, we use the one-vs-all strategy with an RBF-kernel. It consists in building a binary classifier for each class as follows : for each class, a positive label is assigned to the corresponding instances and a negative label is affected to all the remaining instances. The winning class c_{svm} can be chosen based on probability estimates [23] or a simple score maximization :

$$c_{svm} = \underset{1 \leq c \leq N_c}{\operatorname{argmax}} \{ f_{svm}(x, c) \} \quad , \quad f_{svm}(x, c) = \sum_{i=1}^{M_c} \alpha_i^c y_i^c \mathrm{K}(x_i^c, x) + b_c \ , \tag{11}$$

where α_i^c are the optimal Lagrange multipliers of the classifier representing the class c, x_i^c are the support vectors of the class c, y_i^c are the corresponding ± 1 labels, N_c is the number of classes and x is the instance to classify.

Generative PCA Classifier The generative PCA (GPCA) classifier is a simple PCA-based classifier recently used in [15,14]. Given a test descriptor x, GPCA finds the closest class centroid $\mathbb{E}(\{x_c\})$ to x, after ignoring the first D principal variability directions. Let V_c be the linear space generated by the D eigenvectors of the covariance matrix of largest eigenvalues, and $V_c^{\perp}$ its orthogonal complement. The generative PCA classifier uses the projection distance associated to $P_{V_c^{\perp}}$:

$$c_{pca} = \underset{1 \leq c \leq N_c}{\operatorname{argmin}} \, ||P_{V_c^{\perp}} \left(x - \mathbb{E}(\{x_c\}) \right) ||^2. \tag{12}$$

Classification consists in choosing the class c_{pca} with the minimum projection distance.

GPCA-SVM Classifier We propose to combine GPCA and SVMs in one single classifier. The idea behind this combination comes from the observation that SVMs and GPCA often fail on different instances. As a result, a well-established combination of these classifiers should theoretically lead to improved performance. We propose a combination based on the distance between the score separation of each classifier output

$$c_{final} = \begin{cases} c_{svm} & \text{if } f_{svm}(x, c_{svm}) - f_{svm}(x, c_{pca}) \geq th_{svm} \\ c_{pca} & \text{otherwise,} \end{cases} \tag{13}$$

where th_{svm} is a threshold parameter. The score separation gives an idea of SVMs' accuracy to classify a given instance. Another similar approach would be using probability estimates [23] instead of the score. If the measure $f_{svm}(x, c_{svm}) - f_{svm}(x, c_{pca})$ is relatively important, this means that SVMs are quite "confident" about the result. Otherwise, the classifier selects the GPCA result. Determining the best threshold th_{svm} for each instance is an open problem. In this paper, we rather fix a threshold value for each experiment. We generally select a small threshold for small training sets and larger thresholds for larger sets. Even if this strategy is not optimal, experiments show that the combination improves the classification rates as expected.

3.2 Synthetic Training

One important problem in training is coping with the low amount of examples. We propose a simple strategy to artificially add more data to the training set by changing illumination and scale of each instance of the training set. While this idea seems simple, it can have a dramatic impact on the performance as we will see in the next section.

Multi-Illumination Training Given an input image I, multi-illumination training consists in generating other images of the same content of I but with different illumination. There are two illumination cases ; the first one consists in *uniform* changing by image scaling of the form aI, where a is a given scalar. The second case consists in *nonuniform* changing using histogram matching with a set of histograms. The histograms can come from external images, or even from the training set itself (for example by transforming or combining a set of histograms).

Multi-Scale Training Given an input image I, multi-scale training consists simply in generating other images of the same size as I by zooming-in and out. In this paper, we use around 4 generated images, 2 by zooming-in and 2 others by zooming-out.

4 Texture Classification Experiments

We present in this section texture classiffication results conducted on standard public datasets **UIUC** [24,1], **UMD** [25] and **ALOT** [26,27], as well as a comparison with **9** state-of-the-art methods.

Datasets Description The **UIUC** dataset [24,1] is one of the most challenging texture datasets presented so far. It is composed of 25 classes, each class contains 40 grayscale images of size 480 × 640 with strong scale, rotation and viewpoint changes in uncontrolled illumination environment. Some images exhibit also strong non-rigid deformations. Some samples are presented in Figure 4. The **UMD** dataset [25] is similar to **UIUC** with higher resolution images (1280 × 960) but exhibits less non-rigid deformations and stronger illumination changes compared to **UIUC**. To evaluate the proposed method on a large-scale dataset, we choose the **ALOT** dataset [26,27]. It consists of **250** classes, 100 samples each. We use the same setup as the previous multifractal methods [13]: grayscale version with half resolution (768 × 512). The **ALOT** dataset is very challenging as it reprensents a significantly larger number of classes (250) compared to **UIUC** and **UMD** (25) and very strong illumination change (8 levels of illumination). The viewpoint change is however less dramatic compared to **UIUC** and **UMD**.

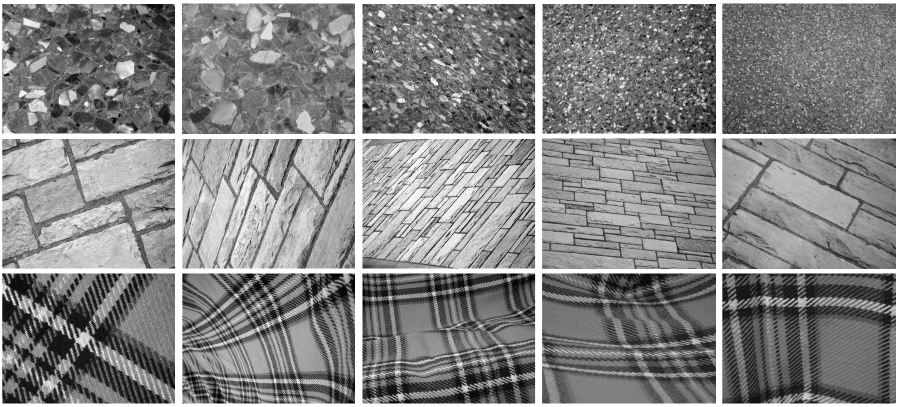

Fig. 4: Texture samples from the **UIUC** dataset [24,1]. Each row represents images from the same class with strong enviromental changes.

Implementation details In order to build a fast texture classification system, we use only two high-pass filtering responses, which results in 3 histograms per image resolution [3]. The number of the image scales is fixed to 5. The filter bank consists in high-pass wavelet filters (Daubechies, Symlets and Gabor). A more robust descriptor can be built by increasing the number of filters and orientations. Filtering can be parallelized for faster processing. While augmenting the number of filters slightly improves classification results, the minimalist setup presented above, coupled with the training strategies introduced in this paper, permits to outperform existing techniques while offering in addition computational efficiency.

Evaluation

We evaluate the proposed system and compare it with state-of-the-art methods for 50 random splits between training and testing. The evaluation consists in three steps :

[3] Except for **ALOT** dataset, we use 3 high-pass responses for a more robust representation.

1. log-histogram vs. box-counting : We evaluate the precision of our log-histogram method and compare it with the box-counting method used in previous methods.
2. Learning efficiency : We compare the proposed GPCA-SVM combination with single GPCA and SVM results and see how the proposed synthetic training strategy improves classification rates.
3. We compare our main results with **9** state-of-the-art results.

log-histogram vs. box-counting In this experiment, we replace the log-histogram step of our approach with the box-counting method widely used in the previous multifractal methods to see if the proposed log-histogram leads to a more accurate bi-Lipschitz invariance. The results are presented in Figure 5. As can be seen, the log-histogram approach leads to higher performance, especially when more data is available. This clearly shows that indeed, the log-histogram leads to a better bi-Lipschitz invariance, as theoretically discussed before. The log-histogram is a simple operation that permits our system to achieve high computational efficiency.

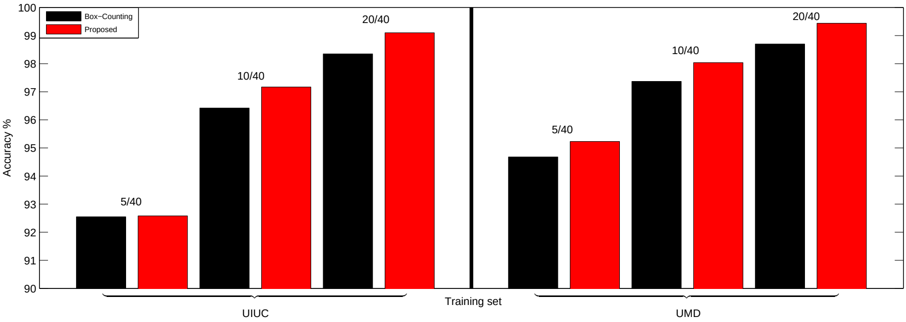

Fig. 5: Comparison between the box-counting method and the proposed log-histogram approach for various dataset training sizes (5, 10 and 20). The proposed approach leads to a more accurate descriptor.

Learning Efficiency In this experiment, we first compare the proposed GPCA-SVM combination with single GPCA and SVM classifiers using the proposed descriptor. Each dataset is presented in the form $D^{x}_{(y)}$ where x is the name of the dataset and y is the training size in number of images. The best results are in bold. As can be seen in Table 1, the GPCA-SVM does indeed improve classification rates. We expect to get even better results with a better strategy to set the threshold parameters th_{svm} as in the proposed experiments, the threshold is fixed for all the instances. Now we compare the results with and without the proposed synthetic training strategy. As can be seen, synthetic training leads to a dramatic improvement. This is a very interesting approach as it increases only the training time. The system can achieve higher recognition accuracy for almost the same computational effiency. For the **UMD** and **ALOT** datasets, we use uniform illumination change with the multiplicative parameter a in the range $[0.9, 0.95, 1.05, 1.1]$. For the **UIUC** dataset, we use the nonuniform illumination change with two histograms. For the multi-scale training, we use only four generated images (two by zooming-in and two other by zooming-out), which increases the training set 9 times in the **UMD** and **UIUC** datasets (no mutli-scale training is used for the **ALOT** dataset).

		$D^{UIUC}_{(5)}$	$D^{UIUC}_{(10)}$	$D^{UIUC}_{(20)}$	$D^{UMD}_{(5)}$	$D^{UMD}_{(10)}$	$D^{UMD}_{(20)}$	$D^{ALOT}_{(10)}$	$D^{ALOT}_{(30)}$	$D^{ALOT}_{(50)}$
Proposed	GPCA	91.15%	97.12%	99.07%	95.07%	97.85%	99.40%	89.30%	98.03%	99.27%
	SVM	91.23%	96.30%	98.47%	94.43%	97.44%	99.25%	88.96%	98.16%	99.14%
	GPCA-SVM	**92.58%**	**97.17%**	**99.10%**	**95.23%**	**98.04%**	**99.44%**	**90.67%**	**98.45%**	**99.34%**
+ Synthetic Train	GPCA	95.84%	98.77%	99.67%	98.02%	99.13%	99.62%	91.54%	98.81%	99.59%
	SVM	95.40%	98.43%	99.46%	97.75%	99.06%	99.72%	92.23%	98.80%	99.51%
	GPCA-SVM	**96.13%**	**98.93%**	**99.78%**	**98.20%**	**99.24%**	**99.79%**	**92.82%**	**99.03%**	**99.64%**

Table 1: Classification rates comparison using GPCA-SVM and synthetic training.

Discussions We compare the proposed method MCMA (Multilayer Convolution - Multifractal Analysis) with **9** state-of-the-art methods for 50 random splits between training and testing, for different training sizes. Results are presented in Table 2. The best results are in bold [4]. As can be seen, the proposed method outperforms the published results on the 3 datasets. Compared to the leading method [14], our system seems to better handle viewpoint change and non-rigid deformations. This is clearly shown in the results on the **UIUC** dataset that exhibits strong enviromental changes. This result can be expected as the scattering method builds invariants on translation, rotation and scale changes, which does not include viewpoint change and non-rigid deformations. Contrary to this, using accurate multifractal statistics, our solution produces descriptors that are invariant to these complex transformations. The proposed system maintains a high performance on the **UMD** dataset. It is worth noting that on this dataset, the images are of high resolution (1280 × 960), which gives an advantage over the **UIUC** dataset. However, we did not use the original resolution, we rather rescale the images to half-size for faster processing. The high accuracy shows that the proposed multifractal method is able to extract robust invariant statistics even on low-resolution images. On the large-scale dataset **ALOT**, the proposed method maintains high performance. Recall that this dataset contains **250** classes with 100 samples each. This is a very challenging dataset that evaluates the extra-class decorrelation of the produced descriptors. A robust descriptor should increase the intra-class correlation, but should also decrease the extra-class correlation and this has be evaluated on a large-scale data set which contains as many different classes as possible. The results on the **ALOT** dataset clearly show a significant performance drop of the leading multifractal method WMFS. The proposed solution in fact outperforms the WMFS method even without synthetic train as can be seen in Table 1. This proves that the proposed descriptor is able to extract a robust invariant representation.

[4] Detailed results with standard deviation can be found in the supplementary material.

	$D^{UIUC}_{(5)}$	$D^{UIUC}_{(10)}$	$D^{UIUC}_{(20)}$	$D^{UMD}_{(5)}$	$D^{UMD}_{(10)}$	$D^{UMD}_{(20)}$	$D^{ALOT}_{(10)}$	$D^{ALOT}_{(30)}$	$D^{ALOT}_{(50)}$
MFS [10]	-	-	92.74%	-	-	93.93%	71.35%	82.57%	85.64%
OTF-MFS [11]	-	-	97.40%	-	-	98.49%	81.04%	93.45%	95.60%
WMFS [13]	93.40%	97.00%	97.62%	93.40%	97.00%	98.68%	82.95%	93.57%	96.94%
VG-Fractal [9]	85.35%	91.64%	95.40%	-	-	96.36%	-	-	-
Varma [28]	-	-	98.76%	-	-	-	-	-	-
Lazebnik [1]	91.12%	94.42%	97.02%	90.71%	94.54%	96.95%	-	-	-
BIF [5]	-	-	98.80%	-	-		-	-	-
SRP [7]	-	-	98.56%	-	-	99.30%	-	-	-
Scattering [14]	93.30%	97.80%	99.40%	96.60%	98.90%	99.70%	-	-	-
MCMA	**96.13%**	**98.93%**	**99.78%**	**98.20%**	**99.24%**	**99.79%**	**92.82%**	**99.03%**	**99.64%**

Table 2: Classification rates on the **UIUC**,**UMD** and **ALOT** datasets.

5 Conclusion

This paper presents a fast and accurate texture classification system. The proposed solution builds a locally invariant representation using a multilayer convolution architecture that performs convolutions with a filter bank, applies a pooling operator to increase the local invariance and repeats the process for various image resolutions. The resulting images are mapped into a stable descriptor via multifractal analysis. We present a new multifractal descriptor that extracts rich texture information from the local singularity exponents. The descriptor is mathematically validated to be invariant to bi-Lipschitz transformations, which includes complex environmental changes. The second part of paper tackles the training part of the recognition system. We propose the GPCA-SVM classifier that combines the generative PCA classifier with the popular kernel SVMs to achieve higher accuracy. In addition, a simple and efficient ”synthetic training” strategy is proposed that consists in synthetically generating more training data by changing illumination and scale of the training instances. Results outperforming the state-of-the-art are obtained and compared with 9 recent methods on 3 challenging public benchmark datasets, while ensuring high computational efficiency.

Acknowledgements

Hicham Badri’s PhD is funded by an INRIA (Direction of Research) CORDI-S grant. He is making a PhD in co-supervision with INRIA and Mohammed V-Agdal University - LRIT, Associated Unit to CNRST (URAC 29).

References

1. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. PAMI **27** (2005) 1265–1278
2. Zhang, J., Marszalek, M., Lazebnik, S., Schmid, C.: Local features and kernels for classification of texture and object categories: A comprehensive study. Int. J. Comput. Vision **73**(2) (June 2007) 213–238

3. Varma, M., Zisserman, A.: A statistical approach to material classification using image patch exemplars. PAMI **31**(11) (November 2009) 2032–2047
4. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. PAMI **24**(7) (July 2002) 971–987
5. Crosier, M., Griffin, L.D.: Texture classification with a dictionary of basic image features. In: CVPR, IEEE Computer Society (2008)
6. Liu, L., Fieguth, P.W.: Texture classification from random features. PAMI **34**(3) (2012) 574–586
7. Liu, L., Fieguth, P.W., Kuang, G., Zha, H.: Sorted random projections for robust texture classification. In: ICCV. (2011) 391–398
8. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA, USA (2001)
9. Varma, M., Garg, R.: Locally invariant fractal features for statistical texture classification. In: CVPR, Rio de Janeiro, Brazil. (October 2007)
10. Xu, Y., Ji, H., Fermuller, C.: A projective invariant for textures. 2006 CVPR **2** (2006) 1932–1939
11. Xu, Y., Huang, S.B., Ji, H., Fermller, C.: Combining powerful local and global statistics for texture description. In: CVPR, IEEE (2009) 573–580
12. Xu, Y., Yang, X., Ling, H., Ji, H.: A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid. In: CVPR. (2010) 161–168
13. Ji, H., Yang, X., Ling, H., Xu, Y.: Wavelet domain multifractal analysis for static and dynamic texture classification. IEEE Transactions on Image Processing **22**(1) (2013) 286–299
14. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: CVPR. (2013)
15. Bruna, J., Mallat, S.: Invariant scattering convolution networks. PAMI **35**(8) (August 2013) 1872–1886
16. Falconer, K.: Techniques in Fractal Geometry. Wiley (1997)
17. Xu, Y., Ji, H., Fermüller, C.: Viewpoint invariant texture description using fractal analysis. Int. J. Comput. Vision **83**(1) (June 2009) 85–100
18. Arneodo, A., Bacry, E., Muzy, J.F.: The thermodynamics of fractals revisited with wavelets. Physica A: Statistical and Theoretical Physics **213**(1-2) (January 1995) 232–275
19. Turiel, A., del Pozo, A.: Reconstructing images from their most singular fractal manifold. IEEE Trans. Img. Proc. **11**(4) (April 2002) 345–350
20. Yahia, H., Turiel, A., Perez-Vicente, C.: Microcanonical multifractal formalism: a geometrical approach to multifractal systems. Part I: singularity analysis. Journal of Physics A: Math. Theor (41) (2008)
21. Turiel, A., Parga, N.: The multifractal structure of contrast changes in natural images: From sharp edges to textures. Neural Computation **12**(4) (2000) 763–793
22. Wendt, H., Roux, S.G., Jaffard, S., Abry, P.: Wavelet leaders and bootstrap for multifractal analysis of images. Signal Process. **89**(6) (June 2009) 1100–1114
23. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27
24. : UIUC : `http://www-cvr.ai.uiuc.edu/ponce_grp/data/`.
25. : UMD : `http://www.cfar.umd.edu/~fer/website-texture/texture.htm`.
26. Burghouts, G.J., Geusebroek, J.M.: Material-specific adaptation of color invariant features. Pattern Recognition Letters **30** (2009) 306–313
27. : ALOT : `http://staff.science.uva.nl/~aloi/public_alot/`.
28. Varma, M.: Learning the discriminative powerinvariance trade-off. In: In ICCV. (2007)