

Adaptive Matrices and Filters for Color Texture Classification

Ioannis Giotis · Kerstin Bunte · Nicolai Petkov · Michael Biehl

Published online: 12 June 2012

© The Author(s) 2012. This article is published with open access at Springerlink.com

Abstract In this paper we introduce an integrative approach towards color texture classification and recognition using a supervised learning framework. Our approach is based on Generalized Learning Vector Quantization (GLVQ), extended by an adaptive distance measure, which is defined in the Fourier domain, and adaptive filter kernels based on Gabor filters. We evaluate the proposed technique on two sets of color texture images and compare results with those other methods achieve. The features and filter kernels learned by GLVQ improve classification accuracy and they are able to generalize much better for data previously unknown to the system.

Keywords Adaptive metric · Adaptive filters · Classification · Color texture analysis · Gabor filters · Learning Vector Quantization

1 Introduction

Texture analysis and classification are topics of interest due to their numerous possible applications, such as medical imaging, industrial quality control and remote sensing. A wide variety of methods for texture analysis has been developed such as co-occurrence matrices [11], Markov random fields [34], autocorrelation methods [24, 29], Gabor

filtering [6, 9, 16, 18, 21, 32] and wavelet decomposition [33]. These methods mostly concern intensity images and since color information is a vector quantity an adaptation to the color domain is not always straightforward. Regarding color texture, the possible approaches can be divided in three categories [25] called parallel, sequential and integrative. In the parallel approach [22, 27] textural features are extracted solely from the luminance plane of an image and are used together with color features. The sequential approach [12] involves a quantization of the color space and subsequently the extraction of statistical features from the indexed images.

The integrative approach [5, 14, 15, 20, 25] is the most popular one and it describes color texture by combining color information with the spatial relationships of image regions within each color channel and between different color channels. The simplest integrative approach would only consist of a gray scale transformation of the input image but in many cases this has been proven insufficient. A very common advance of the integrative approach is based on the opponent-process theory of human color vision that has its roots in neuroscience. Ewald Hering [13] first noted that there are some color combinations that humans are not able to see, such as reddish-green or yellowish-blue, since these colors contrast each other strongly. Hence, he proposed that such color combinations can be the components of one vision mechanism that oppose each other through a process of excitatory and inhibitory responses. A popular application of this theory in computer vision is the Gaussian color model [8].

In this contribution we introduce a novel integrative approach towards color texture classification and recognition based on adaptive filters through supervised learning. The kernels we use are initialized as two-dimensional Gabor filters. A 2D Gabor filter acts as a local band-pass filter and can achieve optimal joint localization both in the spatial and

I. Giotis (✉) · N. Petkov · M. Biehl
Johann Bernoulli Institute for Mathematics and Computer Science, University of Groningen, PO Box 407, 9700 AK, Groningen, The Netherlands
e-mail: i.e.giotis@rug.nl

K. Bunte
CITEC Center of Excellence—Cognitive Interaction Technology, Bielefeld University, Universitaetsstrasse 21-23, 33615, Bielefeld, Germany

frequency domains [4]. Given a set of labeled color images (RGB) for training and a bank of 2D Gabor filter kernels the goal here is to learn a transformation of a color image to a single channel (intensity) image and an optimal adaptation of the kernels such that the responses of the transformed images when filtered with the optimized kernels will yield the best possible classification.

Many signal processing techniques are based on insights or empirical observations from neurophysiology or optical physics. The proposed novel approach incorporates data-driven adaptation of the system, e.g. example based learning. Furthermore, the “family” of filters used in our approach can be substituted, depending on the data domain and the task at hand. As an example we explore in this paper the use of rotation and scale invariant descriptors based on Gabor filter responses [10]. We demonstrate that our approach yields very good generalization ability.

The paper is structured as follows: In Sects. 2 and 3 we present overviews of the existing approaches for color texture analysis and the Learning Vector Quantization algorithm respectively. In Sect. 4 the Color Image Analysis LVQ is explained in detail and Sect. 5 presents experimental results. Finally, in Sect. 6 we draw conclusions.

2 Overview of Existing Approaches

In texture analysis Gabor filter responses and Local Binary Patterns are two very popular types of descriptor that have been extended to color texture via integrative approaches that are using the opponent color model.

Jain et al. [15] proposed an approach that extends the use of features extracted from Gabor filter responses to color texture classification motivated by mechanisms of human vision. For this purpose they compute features from each color channel independently (unichrome features), as well as features that capture the spatial correlation between spectral bands (opponent features). Let h_{imn} be the response of the i -th color channel of a given image when filtered with a Gabor kernel with scale m and orientation n . The unichrome features are defined as the square root of the energy of the Gabor responses:

$$\mu_{imn} = \sqrt{\left(\sum_{x,y} h_{imn}^2(x,y)\right)}.$$

The opponent features are based on the difference of normalized energies between different color channels and scales in the same orientation. The difference of normalized energies is:

$$d_{ijm'mn} = \left(\frac{h_{imn}}{\mu_{imn}} - \frac{h_{jm'n}}{\mu_{jm'n}}\right)$$

thus defining the opponent feature:

$$\psi_{ijm'mn} = \sqrt{\left(\sum_{x,y} d_{ijm'mn}^2(x,y)\right)}.$$

All unichrome and opponent features are concatenated into a single feature vector that is used as a descriptor for the given image. In the following we refer to this technique as Opponent Color Features (OCF).

Local Binary Patterns (LBP) are based on the idea that texture can be described by local spatial patterns and gray scale contrast. The original LBP operator [23] creates labels for the image pixels by thresholding their 3×3 neighborhood with the center value. The pixels with lower intensities than the center pixel are labeled with 0, whereas those with equal or higher intensity values are labeled with 1. The labels are read clockwise as a binary number. This process is repeated for every pixel and the histogram of the 256 possible binary numbers is then used as a texture descriptor. The LBP operator was further extended to use neighborhoods of different sizes [24] using circular neighborhoods and bi-linearly interpolated values at non-integer pixel coordinates. In the following, the notation (P, R) is used to denote pixel neighborhoods formed by P sampling points on a circle of radius R . Another extension to the original operator is the definition of the so called uniform patterns, which can be used to reduce the length of the feature vector and implement a simple rotation-invariant descriptor. This extension was inspired by the fact that some binary patterns occur more commonly in texture images than others. A local binary pattern is called uniform if it contains at most two transitions from 0 to 1 or vice versa when it is traversed circularly. Ojala et al. [24] noticed in their experiments that uniform patterns account for a little less than 90 % of all patterns when using a $(8, 1)$ neighborhood and for around 70 % with a $(16, 2)$ neighborhood. After the LBP labeled image $f_l(x, y)$ has been obtained, the descriptor is defined as:

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\}, \quad i = 0, \dots, n-1.$$

The Color LBP extension [28] is based on the ability to take the local threshold (neighborhood center) from n different color channels. The neighborhood to be thresholded can also be taken from these channels, which makes up a total of n^2 different combinations. The n^2 histogram descriptors are then concatenated into a single feature vector.

3 Review of the (Generalized Matrix) Learning Vector Quantization

Learning Vector Quantization (LVQ) is a supervised prototype-based classification method [17]. The training is based

on data points $\mathbf{x}^i \in \mathbb{R}^D$ and their corresponding label information $y^i \in \{1, \dots, C\}$, where D denotes the dimension of the feature vectors and C the number of classes. A set of prototypes is characterized by their location in the feature space $\mathbf{w}^i \in \mathbb{R}^D$ and the respective class label $c(\mathbf{w}^i) \in \{1, \dots, C\}$. Classification is implemented as a winner-takes-all scheme. For this purpose, a possibly parameterized dissimilarity measure d^Ω is defined, where Ω specifies the metric parameters which can be adapted during training. Given $d^\Omega(\mathbf{x}, \mathbf{w})$, any data point \mathbf{x} is assigned to the class label $c(\mathbf{w}^i)$ of the closest prototype \mathbf{w}^i with $d^\Omega(\mathbf{x}, \mathbf{w}^i) \leq d^\Omega(\mathbf{x}, \mathbf{w}^j)$ for all $j \neq i$. The position of the closest (“winner”) prototype in the feature space is then adapted according to a learning rule, i.e. \mathbf{w}^i is moved closer to \mathbf{x} if the data point is correctly classified and moved away from \mathbf{x} if otherwise. The number of prototypes used to represent a class can be chosen by the user according to the nature of the data and the task at hand. The typical number of prototypes assigned to each class varies from 1 to 5.

A training scheme called Generalized LVQ (GLVQ) [30] is derived as a minimization of the cost function:

$$f_c(d^\Omega, J, K) = \sum_i \Phi \left(\frac{d^\Omega(\mathbf{x}^i, \mathbf{w}^J) - d^\Omega(\mathbf{x}^i, \mathbf{w}^K)}{d^\Omega(\mathbf{x}^i, \mathbf{w}^J) + d^\Omega(\mathbf{x}^i, \mathbf{w}^K)} \right) \quad (1)$$

where the quantities

$$d^\Omega(\mathbf{x}^i, \mathbf{w}^J) \quad \text{with } c(\mathbf{w}^J) = c(\mathbf{x}^i) \quad (2)$$

$$d^\Omega(\mathbf{x}^i, \mathbf{w}^K) \quad \text{with } c(\mathbf{w}^K) \neq c(\mathbf{x}^i) \quad (3)$$

correspond to the distances of the feature vector \mathbf{x}^i from the respective closest correct prototype \mathbf{w}^J and the closest wrong prototype \mathbf{w}^K . Φ must be a monotonic function and throughout the following the identity $\Phi(x) = x$ is used.

Generalized Matrix Learning Vector Quantization (GM-LVQ) is an extension of the original algorithm with adaptive dissimilarity measure based on the quadratic form:

$$d^\Omega(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^\top \Omega^\top \Omega (\mathbf{x} - \mathbf{w}) \quad (4)$$

The matrix $\Lambda = \Omega^\top \Omega$ is assumed to be positive (semi-)definite. Hence the measure corresponds to a (squared) Euclidean distance in an appropriately transformed space

$$d^\Omega(\mathbf{x}, \mathbf{w}) = [\Omega(\mathbf{x} - \mathbf{w})]^2 \quad (5)$$

Specific restrictions may be imposed on the transformation $\Omega \in \mathbb{R}^{M \times D}$ with $M \leq D$ without loss of generality. For $M < D$, Ω transforms the D -dimensional data into a lower M -dimensional space. This variant is referred to as Limited Rank Matrix LVQ (LiRaM LVQ) and explained in [1, 2]. The original algorithm follows a stochastic gradient descent for the optimization of the cost function (Eq. (1)). The gradients are evaluated with respect to the contribution of single instances \mathbf{x}^i , which are presented in random order and

sequentially during training. The algorithm has been introduced and discussed in [31] and will be modified in the subsequent sections.

4 Color Image Analysis Learning Vector Quantization

In this contribution we present an extension of the GMLVQ concept, that is especially designed for color texture analysis. We use the same cost function, Eq. (1), as in the original GMLVQ algorithm and follow a stochastic gradient descent procedure where the samples \mathbf{x}^i of the training set are sequentially presented and the parameters accordingly updated. We will refer to this algorithm as Color Image Analysis LVQ (CIA-LVQ) and to one sweep through the training set as one epoch E .

Let \mathbf{D} be a data set of color images of a priori known size ($p \times p$) that belong to C different classes and a bank of filter kernels \mathbf{G} , initialized as a sum of Gabor filters with different scales and orientations. The goal is to learn one or more matrices Ω_k that transform the color images into a single-channel, “intensity” image, a set of optimized kernels \widehat{G}_k and a set of prototypes \mathbf{w}^k such that the filter responses of the transformed images will yield the best possible classification. In addition, we use an adaptation of the learning rates that allows the system to be less dependent on their initial values.

We use for both the filter kernels and the images their representation in the Fourier domain. The image data are vectorized thus resulting in a data set of complex vectors $\mathbf{x}^i \in \mathbb{C}^N$, where $N = p \cdot p \cdot 3$, with p denoting the image patch size. These vectors are transformed by $\Omega_k \in \mathbb{C}^{M \times N}$, where $M = p \cdot p$. The transformation $\Omega_k \in \mathbb{C}^{M \times N}$ can be considered as the equivalent of a color to gray scale image transformation, with k referring to the index of a prototype \mathbf{w}^k or the index of its class label for class-wise transformations. Subsequently, the transformed image data are filtered with every kernel $G_l \in \mathbf{G}$ and the l responses are summed up. The filter kernels are also represented as complex vectors $G_l \in \mathbb{C}^M$. The general form of the descriptor of an individual image is denoted as:

$$\mathbf{r}_k^i = \mathbf{x}^i \Omega_k^\top * \sum_l G_l \quad (6)$$

where $*$ denotes the convolution. Each such descriptor is associated with a label $y^i \in 1, 2, \dots, C$.

Note that, Eq. (6) describes only one convolution with a sum of kernels $\widehat{G} = \sum_l G_l$. At this point, this is not of conceptual value. Since the algebraic property of distributivity holds for the operation of convolution in the Fourier domain, Eq. (6) yields a result identical to what is described above and can be simplified as:

$$\mathbf{r}_k^i = \mathbf{x}^i \Omega_k^\top * \widehat{G} \quad (7)$$

This obviously offers a gain in processing time, especially for larger filter banks and also simplifies the optimization process. In the following we optimize the sum of kernels \widehat{G} .

We define the dissimilarity measure as:

$$d_{\widehat{G}_k}^{\Omega_k}(\mathbf{x}^i, \mathbf{w}^k) = \left\| |\mathbf{r}_k^i|^2 - |\mathbf{w}^k|^2 \right\|^2 \quad (8)$$

which corresponds to the difference of magnitudes between a prototype and an image descriptor. In this fashion we ensure that two images containing the same texture pattern are considered similar, independent of the position within the image where this pattern occurs.

4.1 Explicit Form of the Learning Rules

The learning rules of CIA-LVQ can be derived from the dissimilarity measure as presented in Eq. (8) by taking the derivatives with respect to the parameters \mathbf{w}^k , Ω_k and \widehat{G}_k . The parameter updates read as follows:

$$\mathbf{w}^L = \mathbf{w}^L - \alpha \Delta \mathbf{w}^L \quad (9)$$

$$\Omega_L = \Omega_L - \epsilon \Delta \Omega_L \quad (10)$$

$$\widehat{G}_L = \widehat{G}_L - \eta \Delta \widehat{G}_L \quad (11)$$

where

$$\Delta \mathbf{w}^L = \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Re(\mathbf{w}^L)} + i \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Im(\mathbf{w}^L)} \quad (12)$$

$$\Delta \Omega_L = \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Re(\Omega_L)} + i \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Im(\Omega_L)} \quad (13)$$

$$\Delta \widehat{G}_L = \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Re(\widehat{G}_L)} + i \frac{\partial f_c(d_{\widehat{G}_J}^{\Omega_J}, d_{\widehat{G}_K}^{\Omega_K}, J, K)}{\partial \Im(\widehat{G}_L)} \quad (14)$$

In Eqs. (9)–(14) $L \in \{J, K\}$ and α , ϵ and η are the learning rates for the prototypes, the transformation matrix and the kernel used for filtering respectively.

The derivatives with respect to the closest correct \mathbf{w}^J and closest wrong prototype \mathbf{w}^K together with the corresponding matrices Ω_J , Ω_K and the filter kernels \widehat{G}_J , \widehat{G}_K for the given training data point \mathbf{x}^i read:

$$\Delta \mathbf{w}^L = -4 \cdot \gamma^L \left[(|\mathbf{r}_L^i|^2 - |(\mathbf{w}^L)|^2) \cdot (\mathbf{w}^L) \right] \quad (15)$$

$$\Delta \Omega_L = \gamma^L \left[4(|\mathbf{r}_L^i|^2 - |(\mathbf{w}^L)|^2) \cdot |\widehat{G}_L|^2 \cdot (\mathbf{x}^i \Omega_L^T) \mathbf{x}^{i*} \right] \quad (16)$$

$$\Delta \widehat{G}_L = \gamma^L \left[4(|\mathbf{r}_L^i|^2 - |(\mathbf{w}^L)|^2) \cdot |\mathbf{x}^i \Omega_L^T|^2 \right] * (\widehat{G}_L) \quad (17)$$

with $*$ denoting the complex conjugate and

$$\gamma^J = \frac{2 \cdot d_{\widehat{G}_K}^{\Omega_K}(\mathbf{x}^i, \mathbf{w}^K)}{(d_{\widehat{G}_J}^{\Omega_J}(\mathbf{x}^i, \mathbf{w}^J) + d_{\widehat{G}_K}^{\Omega_K}(\mathbf{x}^i, \mathbf{w}^K))^2} \quad (18)$$

$$\gamma^K = \frac{-2 \cdot d_{\widehat{G}_J}^{\Omega_J}(\mathbf{x}^i, \mathbf{w}^J)}{(d_{\widehat{G}_J}^{\Omega_J}(\mathbf{x}^i, \mathbf{w}^J) + d_{\widehat{G}_K}^{\Omega_K}(\mathbf{x}^i, \mathbf{w}^K))^2}. \quad (19)$$

Note, that since we are working with complex values we have to take all derivatives with respect to the real and imaginary parts respectively.

4.2 Adaptation of the Learning Rates

Steepest descent methods rely upon the choice of the suitable magnitude for the update step (learning rate). Very small steps usually only slow down convergence, whereas very large steps might result in oscillatory or divergent behavior. In the case of CIA-LVQ the update steps are denoted as α , ϵ and η and the issue of choosing their values is addressed by considering way-point averages over a number of latest iteration steps together with an efficient step size adaptation. This technique is being discussed in [26] for normalized gradients, but in CIA-LVQ we use its basic principles without the normalization.

The general form of the update of a parameter \mathbf{x} is an iterative process with an initial learning rate value ψ_0 and an initial parameter value \mathbf{x}_0 . At every iteration step the cost function $f_c(\mathbf{x}_j)$ is computed. At first we perform $k > 1$ unaltered gradient steps as follows:

$$\mathbf{x}_{j+1} = \mathbf{x}_j - \psi_j \Delta \mathbf{x}_j \quad (20)$$

for $j = 0, 1, \dots, k-1$ with $\psi_j = \psi_0$. Consequently, apart from the current gradient step $\tilde{\mathbf{x}}_{t+1}$ we also compute the way-point average of the previous k steps:

$$\hat{\mathbf{x}}_{t+1} = \frac{1}{k} \sum_{i=0}^{k-1} \mathbf{x}_{t-i} \quad (21)$$

We determine the new position of the parameter \mathbf{x}_{t+1} and the new step size ψ_{t+1} as:

$$\mathbf{x}_{t+1} = \begin{cases} \tilde{\mathbf{x}}_{t+1} & \text{if } f_c(\tilde{\mathbf{x}}_{t+1}) \leq f_c(\hat{\mathbf{x}}_{t+1}), \\ \hat{\mathbf{x}}_{t+1} & \text{otherwise} \end{cases} \quad (22)$$

$$\psi_{t+1} = \begin{cases} \psi_t & \text{if } f_c(\tilde{\mathbf{x}}_{t+1}) \leq f_c(\hat{\mathbf{x}}_{t+1}), \\ \beta \cdot \psi_t & \text{otherwise, with } \beta < 1 \end{cases} \quad (23)$$

As long as a simple gradient descent step yields a position for the parameter \mathbf{x} that results in lower costs than the average of the k latest positions of \mathbf{x} , the iterative process remains unaltered. On the other hand, $f_c(\tilde{\mathbf{x}}_{t+1}) > f_c(\hat{\mathbf{x}}_{t+1})$ indicates that the step size is too large and should be reduced by a factor β .

In the next section we experiment with the algorithm and show its use in practice.

5 Experiments

In order to evaluate the usefulness of the proposed algorithm, we perform classification on patches of pictures taken from the VisTex [3] and the KTH-TIPS [7] databases. From the VisTex database we use 29 color images with size 128×128 pixels from the groups Bark, Brick, Fabric and Food. The KTH-TIPS set is used in its original form and consists of 810 color images with size 200×200 pixels from 10 different classes: Sandpaper, Aluminium Foil, Sponge, Styrofoam, Corduroy, Linen, Brown Bread, Cotton, Orange Peel and Cracker. Although in texture classification literature every image is often considered as a different class, here we distinguish into four and ten different classes respectively, which are equivalent to the conceptual groups that the images belong to. Despite its increased difficulty, this classification task allows us to better demonstrate the ability of CIA-LVQ to describe general characteristics of real-world texture patterns.

We split both data sets in two subsets. One subset is used for training whereas the other is never seen during training and we use it for evaluation. Figures 1 and 2 depict the training and evaluation images from the VisTex database. Fig-

ures 3 and 4 depict examples of training and evaluation images respectively from the KTH-TIPS database.

For our experiments we draw 15×15 patches randomly from each image. The training subsets of images are further



Fig. 1 Images, used to provide patches for training and test (VisTex)

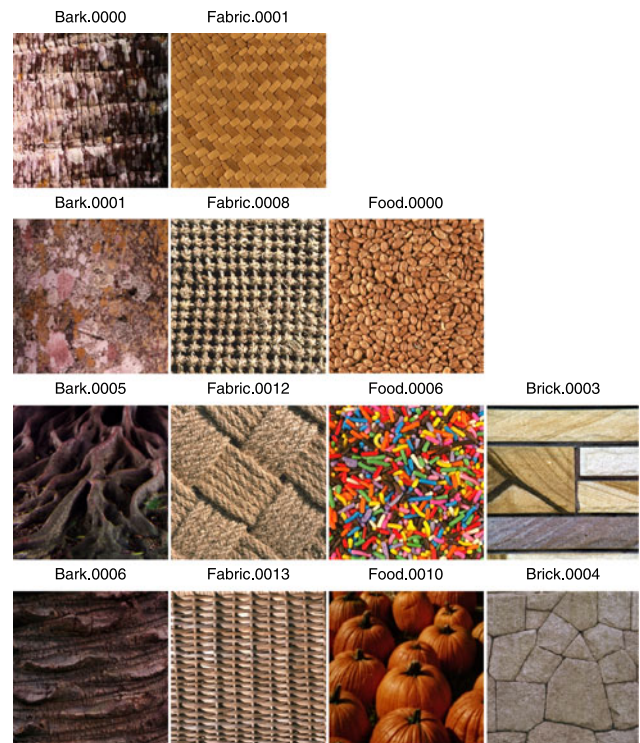


Fig. 2 Images, used to provide patches for evaluation (VisTex)



Fig. 3 Images, used to provide patches for training and test (KTH-TIPS)



Fig. 4 Images, used to provide patches for evaluation (KTH-TIPS)

divided in training and test sets of patches. The VisTex training subset consists of 200 patches per image. We use 150 patches per image (2400 data points) for training and test the performance of CIA-LVQ on the remaining 50 patches per image (800 data points). With respect to the KTH-TIPS training subset we draw 9 patches per image and use 6 for training (3240 data points) and the remaining 3 (1640 data points) for testing. The test sets may contain patches which partially overlap with those used for training. Therefore we use the images in Figs. 2 and 4 in order to create evaluation sets that have never been seen in the training process and thus better demonstrate the generalization ability of the proposed approach. The evaluation sets consist of 50 and 6 randomly drawn patches per image for VisTex and KTH-TIPS respectively.

A note is due here to the nature of the filters used for initialization. A 2D Gabor filter is defined as a Gaussian kernel function modulated by a sinusoidal plane wave. All filter kernels can be generated from one basic wavelet by dilation and rotation. In these experiments we initialize the adaptive filter banks as follows: Every bank consists of 16 Gabor filters of bandwidth equal to 1 at eight orientations $\theta = 0, 22.5, 45, 67.5, 90, 112.5, 135$ and 157.5 degrees and two scales (wavelengths) varying by one octave $\lambda = \{5, 5\sqrt{2}\}$. These scales ensure that the Gabor function yields an adequate number of visible parallel excitatory and inhibitory stripe zones. Dependent on the patch size and the nature of the data at hand different scales might be more suitable. We set the phase offset $\phi = 0$ and the aspect ratio $\gamma = 1$ for all filters. In this way we create center-on symmetric filters with circular support.

We run the localized version of CIA-LVQ with matrices Ω_k initialized with the identity matrix and 4 prototypes per class for $E = 300$ epochs. The prototypes are initialized as the mean of the corresponding class. Regarding VisTex the training error is 5.75 % and the error on the test set 15 %. For the KTH-TIPS data set CIA-LVQ reaches training and test errors of 15.4 % and 22.8 % respectively.

We use the same data sets and the same filter banks to compare with the Gabor-based Opponent Color Features (OCF) [15], the Color Local Binary Patterns (Color LBP) [28] and the common approach of deriving textural information only from the luminance plane of images [5]. The luminance approach is considered to often outperform combined color and texture features [19]. We implement this approach with a RGB to gray (RGB2G) transformation, which builds intensity values by a weighted sum of the color components of every pixel:

$$I_{(x,y)} = 0.2989 \cdot R_{(x,y)} + 0.587 \cdot G_{(x,y)} + 0.114 \cdot B_{(x,y)} \quad (24)$$

We again vectorize all patches s and in this case the image patch descriptor is given by

$$\mathbf{r}_2(s) = s * \sum_l \mathbf{G}^l. \quad (25)$$

For OCF we use a k -nearest neighbors (k -NN) classification scheme with precisely the set of features and the dissimilarity measure suggested by the authors of [15], whereas for the Color LBP we use rotation-invariant uniform LBP histograms in (8, 1) neighborhoods and the Euclidean distance in an k -NN scheme. We choose the size of the neighborhood in relation to the patch size and the dimensions of the feature vectors created. With respect to the RGB2G approach we use the k -NN scheme with a dissimilarity measure similar to Eq. (8):

$$d_G(\mathbf{x}^i, \mathbf{x}^j) = \left\| \left| \mathbf{r}_2(\mathbf{x}^i) \right|^2 - \left| \mathbf{r}_2(\mathbf{x}^j) \right|^2 \right\|^2. \quad (26)$$

Regarding all k -NN schemes we cross-validate the number of nearest neighbors using the values $k = 1, 3, \dots, 15$ on the testing image patches from the training subsets. The optimal k obtained is then used for experimenting on the previously unseen evaluation image patches. Ties are solved by defaulting to the 1-NN classifier.

5.1 Comparisons on the VisTex Data Set

The k -NN scheme shows a test error of 9.1 % based on the OCF ($k = 3$), 2 % based on the Color LBP ($k = 1$) and 25.8 % based on the RGB2G transformation ($k = 1$), but the most interesting comparison relies on the evaluation set which displays the generalization ability of each method. Here the k -NN scheme produces much higher error rates of 35.2 %, 25.2 % and 50 % for OCF, Color LBP and RGB2G

Table 1 Confusion matrices for the VisTex evaluation set

| CIA-LVQ: | | | | | |
|--|-------|-------|--------|-------|----------|
| | Bark | Brick | Fabric | Food | Σ |
| Bark | 179 | 2 | 23 | 4 | 208 |
| Brick | 5 | 85 | 1 | 2 | 93 |
| Fabric | 2 | 13 | 176 | 19 | 210 |
| Food | 14 | 0 | 0 | 125 | 139 |
| Σ | 200 | 100 | 200 | 150 | 650 |
| Class-wise accuracy of estimation in % | | | | | |
| | 89.50 | 85.00 | 88.00 | 83.33 | |
| OCF: | | | | | |
| | Bark | Brick | Fabric | Food | Σ |
| Bark | 111 | 10 | 35 | 36 | 192 |
| Brick | 70 | 78 | 10 | 26 | 184 |
| Fabric | 16 | 12 | 155 | 11 | 194 |
| Food | 3 | 0 | 0 | 77 | 80 |
| Σ | 200 | 100 | 200 | 150 | 650 |
| Class-wise accuracy of estimation in % | | | | | |
| | 55.50 | 78.00 | 77.50 | 51.33 | |
| Color LBP: | | | | | |
| | Bark | Brick | Fabric | Food | Σ |
| Bark | 152 | 24 | 2 | 6 | 174 |
| Brick | 21 | 56 | 12 | 1 | 178 |
| Fabric | 2 | 12 | 138 | 3 | 127 |
| Food | 25 | 8 | 4 | 140 | 181 |
| Σ | 200 | 100 | 200 | 150 | 650 |
| Class-wise accuracy of estimation in % | | | | | |
| | 76.00 | 56.00 | 69.00 | 93.33 | |
| RGB2G: | | | | | |
| | Bark | Brick | Fabric | Food | Σ |
| Bark | 79 | 12 | 38 | 38 | 167 |
| Brick | 64 | 62 | 34 | 28 | 188 |
| Fabric | 16 | 15 | 113 | 13 | 157 |
| Food | 41 | 11 | 15 | 71 | 138 |
| Σ | 200 | 100 | 200 | 150 | 650 |
| Class-wise accuracy of estimation in % | | | | | |
| | 39.50 | 62.00 | 56.50 | 47.33 | |

respectively, while the CIA-LVQ has an error of 13.1 %, in the same order of magnitude as for the test patches. Table 1 presents in detail the confusion matrices and classwise accuracies of all methods for the evaluation set.

CIA-LVQ consistently outperforms all other methods and displays remarkable ability to generalize for previously unknown data. The magnitude of the prototypes, which classify the evaluation set are shown in Fig. 5. Additionally we show some example patches from the evaluation set, which are classified correctly together with their descriptors in Fig. 6 and some examples of wrongly classified patches in Fig. 7. Finally, Fig. 8 depicts in the spatial domain the optimized sums of kernels that are used together with the corresponding prototypes in order to classify the evaluation set. The accuracy rates of the proposed approach don't vary a lot among the different classes. However, Brick and Food are the most difficult to classify using a small patch size due to the large size of the texture patterns and the possibly low contrast respectively. Therefore, Color LBP being invariant to monotonic contrast changes outperforms CIA-LVQ for the class Food.

5.2 Comparisons on the KTH-TIPS Database

The *k*-NN scheme shows a test error of 41.7 % based on the OCF (*k* = 13), 26.4 % based on the Color LBP (*k* = 11) and of 52.7 % based on the RGB2G transformation (*k* = 11), which are all higher than what CIA-LVQ can achieve. On the evaluation set the superior performance of the proposed technique is further clarified. The *k*-NN scheme reaches error rates of 46.4 %, 35.6 % and 58.4 % for OCF, Color LBP and RGB2G respectively, while the CIA-LVQ has an error of 20.3 %, again in the same order of magnitude as for the test patches. Table 2 presents in detail the confusion matrices and classwise accuracies of all methods for the evaluation set of the KTH-TIPS database.

CIA-LVQ is outperformed only for the Corduroy class by all three methods that we compare with, while Color LBP achieves better results for Aluminium Foil, Brown Bread and Cotton as well. The prototypes, which classify the evaluation set of KTH-TIPS are shown in Fig. 9, together with examples of correctly (Fig. 10) and wrongly (Fig. 11) classified patches and their corresponding descriptors. The optimized sums of kernels that are used are shown in Fig. 12. The classes Corduroy, Brown Bread and Cotton are both characterized from nuances of brown color and very diverse patterns as well as the class Sponge. Therefore, the former two are often mistaken for one another or Sponge from CIA-LVQ. The same occurs also between Cotton and Linen that are dominated by very similar colors and often low contrast. Finally, the performance of CIA-LVQ with regard to the class Aluminium Foil is mostly due to the combination of large textures and the small patch size.

6 Conclusion and Outlook

In this contribution we propose a prototype based framework for color texture classification. As an example we ini-

Fig. 5 Plots of the optimized prototypes $|\mathbf{w}^L|$ actively used to classify the data in the evaluation set of the VisTex database. The names consist of the corresponding class name and the index number (1–4) of the prototype

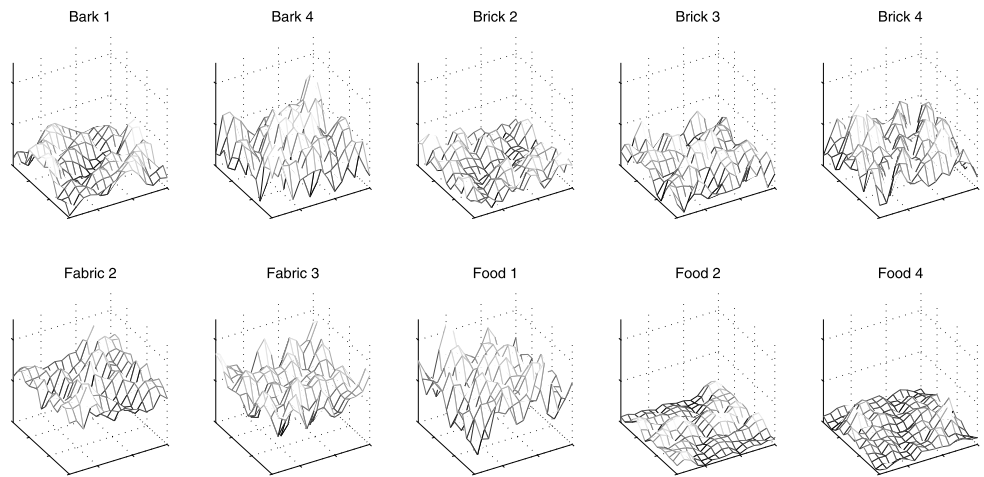


Fig. 6 Plots of the descriptors $|\mathbf{r}_k|$ of some correctly classified image patches from the evaluation set of VisTex database

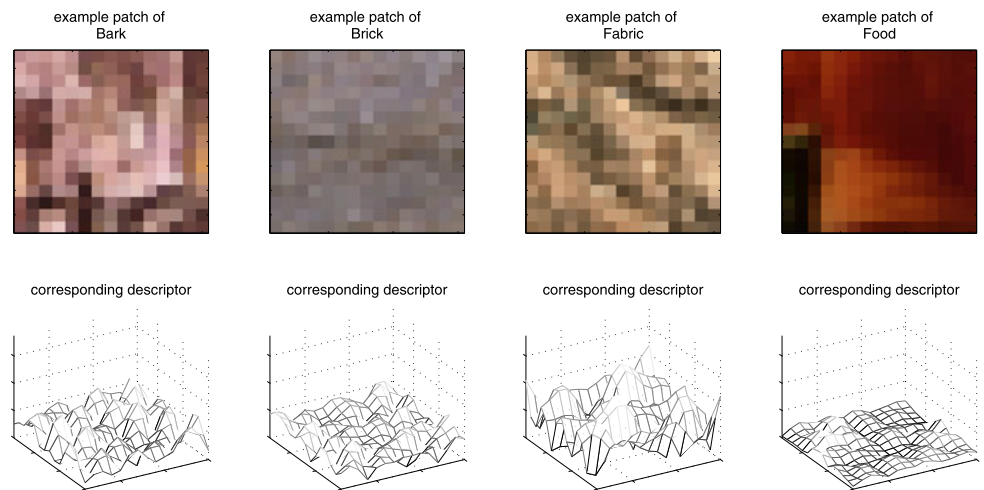
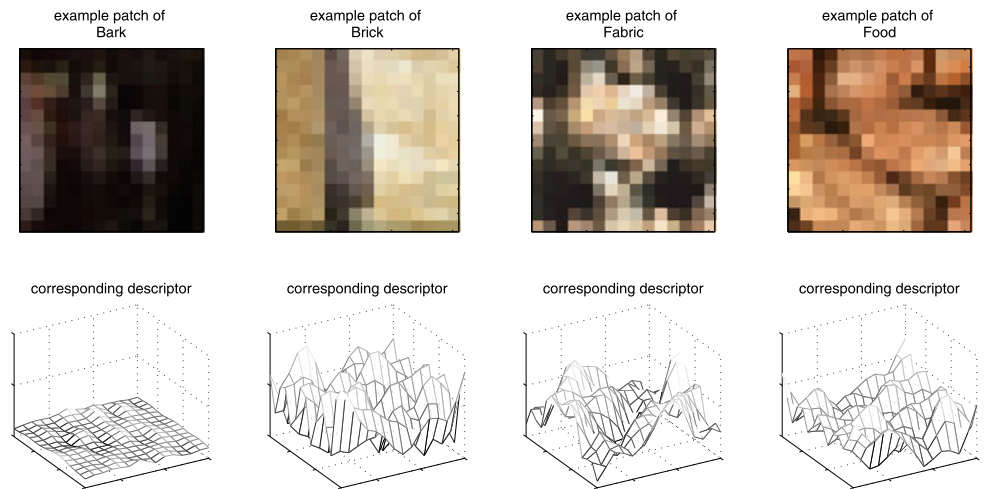


Fig. 7 Plots of the descriptors $|\mathbf{r}_k|$ of some wrongly classified image patches from the evaluation set of VisTex database



tialize the system with Gabor filters and classify color texture patterns in 15×15 patches randomly drawn from images of two public data sets. The results show that CIA-LVQ can learn typical texture patterns with very good generalization, even from relatively small patches and filter banks and it consistently outperforms state of the art techniques used

for color texture analysis. It is also of conceptual value that this LVQ adaptation is suitable for learning in the complex number domain.

The resulting filter kernels may not strictly conform to the notion of Gabor filters, they preserve however the important property of symmetric and periodic excitatory and inhibitory

Fig. 8 Plots of the final form of filter kernels actively used to classify the evaluation set of the VisTex database. The filter kernels have been locally adapted during training. The names consist of the corresponding class name and the index number (1–4) of the kernel

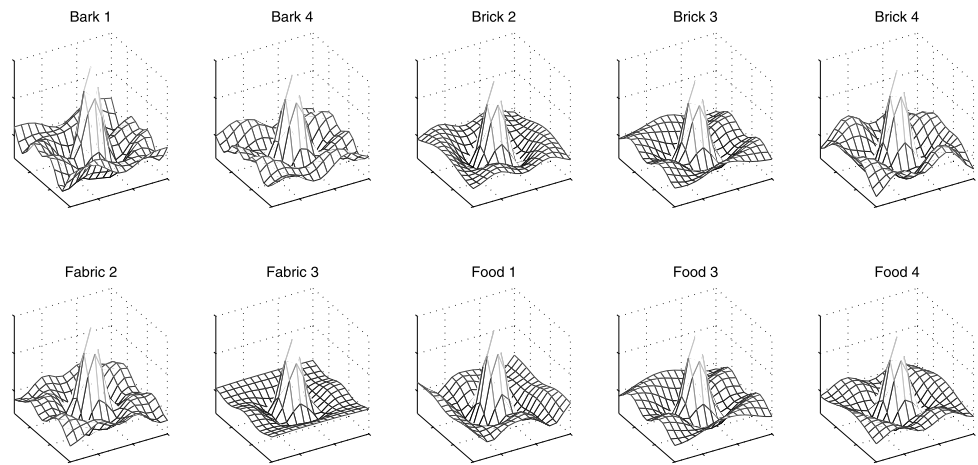


Table 2 Confusion matrices for the KTH-TIPS evaluation set CIA-LVQ:

| | S/paper | Al. Foil | Sponge | Styrofoam | Corduroy | Linen | Br. Bread | Cotton | Or. Peel | Cracker | Σ |
|-----------|---------|----------|--------|-----------|----------|-------|-----------|--------|----------|---------|----------|
| S/paper | 129 | 0 | 2 | 0 | 0 | 4 | 0 | 9 | 0 | 0 | 144 |
| Al. Foil | 0 | 123 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 123 |
| Sponge | 0 | 0 | 107 | 0 | 41 | 0 | 30 | 0 | 0 | 0 | 178 |
| Styrofoam | 0 | 4 | 0 | 133 | 0 | 0 | 0 | 0 | 2 | 0 | 139 |
| Corduroy | 0 | 0 | 22 | 0 | 91 | 0 | 32 | 0 | 0 | 0 | 145 |
| Linen | 0 | 0 | 0 | 2 | 0 | 108 | 0 | 54 | 0 | 0 | 164 |
| Br. Bread | 0 | 0 | 3 | 0 | 3 | 0 | 51 | 0 | 0 | 1 | 58 |
| Cotton | 1 | 7 | 0 | 0 | 0 | 23 | 0 | 70 | 2 | 0 | 103 |
| Or. Peel | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 131 | 1 | 132 |
| Cracker | 5 | 1 | 1 | 0 | 0 | 0 | 22 | 2 | 0 | 133 | 164 |
| Σ | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 1350 |

Class-wise accuracy of estimation in %

| | | | | | | | | | | |
|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 95.56 | 91.11 | 79.26 | 98.52 | 67.41 | 80.00 | 37.78 | 51.85 | 97.04 | 98.52 |
|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|

OCF:

| | S/paper | Al. Foil | Sponge | Styrofoam | Corduroy | Linen | Br. Bread | Cotton | Or. Peel | Cracker | Σ |
|-----------|---------|----------|--------|-----------|----------|-------|-----------|--------|----------|---------|----------|
| S/paper | 34 | 0 | 21 | 26 | 0 | 2 | 7 | 10 | 2 | 7 | 109 |
| Al. Foil | 0 | 112 | 1 | 1 | 2 | 1 | 7 | 4 | 0 | 6 | 134 |
| Sponge | 30 | 0 | 43 | 10 | 8 | 3 | 17 | 4 | 13 | 12 | 140 |
| Styrofoam | 28 | 0 | 15 | 61 | 3 | 17 | 7 | 13 | 0 | 7 | 151 |
| Corduroy | 7 | 4 | 7 | 6 | 97 | 2 | 3 | 22 | 0 | 5 | 153 |
| Linen | 2 | 0 | 0 | 2 | 1 | 91 | 4 | 10 | 2 | 3 | 115 |
| Br. Bread | 16 | 6 | 39 | 13 | 8 | 4 | 58 | 5 | 7 | 33 | 189 |
| Cotton | 3 | 0 | 1 | 5 | 1 | 3 | 1 | 61 | 6 | 0 | 81 |
| Or. Peel | 7 | 0 | 1 | 0 | 1 | 1 | 1 | 4 | 105 | 1 | 121 |
| Cracker | 8 | 13 | 7 | 11 | 14 | 11 | 30 | 2 | 0 | 61 | 157 |
| Σ | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 1350 |

Class-wise accuracy of estimation in %

| | | | | | | | | | | |
|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | 25.19 | 82.96 | 31.85 | 45.19 | 71.85 | 67.41 | 42.96 | 45.19 | 77.78 | 45.19 |
|--|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|

Table 2 (Continued)

Color LBP:

| | S/paper | Al. Foil | Sponge | Styrofoam | Corduroy | Linen | Br. Bread | Cotton | Or. Peel | Cracker | Σ |
|--|---------|----------|--------|-----------|----------|-------|-----------|--------|----------|---------|----------|
| S/paper | 66 | 0 | 27 | 1 | 11 | 4 | 8 | 6 | 0 | 25 | 148 |
| Al. Foil | 0 | 134 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 134 |
| Sponge | 28 | 0 | 65 | 1 | 5 | 0 | 14 | 1 | 28 | 26 | 168 |
| Styrofoam | 0 | 0 | 0 | 126 | 0 | 30 | 0 | 6 | 0 | 2 | 164 |
| Corduroy | 2 | 0 | 8 | 0 | 109 | 1 | 15 | 5 | 0 | 5 | 145 |
| Linen | 0 | 1 | 0 | 1 | 0 | 69 | 0 | 36 | 0 | 0 | 107 |
| Br. Bread | 13 | 0 | 20 | 1 | 6 | 7 | 86 | 1 | 6 | 27 | 167 |
| Cotton | 0 | 0 | 0 | 2 | 1 | 20 | 0 | 72 | 5 | 3 | 103 |
| Or. Peel | 5 | 0 | 15 | 0 | 2 | 0 | 1 | 0 | 95 | 0 | 118 |
| Cracker | 21 | 0 | 0 | 3 | 1 | 4 | 11 | 8 | 1 | 47 | 96 |
| Σ | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 1350 |
| Class-wise accuracy of estimation in % | | | | | | | | | | | |
| | 48.89 | 99.26 | 48.15 | 93.33 | 80.74 | 51.11 | 63.70 | 53.33 | 70.37 | 34.81 | |

RGB2G:

| | S/paper | Al. Foil | Sponge | Styrofoam | Corduroy | Linen | Br. Bread | Cotton | Or. Peel | Cracker | Σ |
|--|---------|----------|--------|-----------|----------|-------|-----------|--------|----------|---------|----------|
| S/paper | 26 | 1 | 17 | 38 | 2 | 11 | 9 | 7 | 24 | 5 | 140 |
| Al. Foil | 1 | 81 | 0 | 0 | 0 | 1 | 3 | 0 | 0 | 8 | 94 |
| Sponge | 22 | 2 | 33 | 13 | 5 | 11 | 16 | 5 | 26 | 14 | 147 |
| Styrofoam | 37 | 1 | 12 | 45 | 0 | 11 | 1 | 8 | 8 | 13 | 136 |
| Corduroy | 0 | 15 | 8 | 1 | 108 | 6 | 9 | 17 | 3 | 7 | 174 |
| Linen | 1 | 5 | 1 | 4 | 3 | 56 | 10 | 2 | 1 | 6 | 89 |
| Br. Bread | 10 | 6 | 22 | 7 | 3 | 9 | 50 | 2 | 10 | 37 | 156 |
| Cotton | 11 | 2 | 4 | 3 | 5 | 12 | 3 | 77 | 12 | 4 | 133 |
| Or. Peel | 24 | 1 | 32 | 19 | 3 | 10 | 7 | 16 | 51 | 6 | 169 |
| Cracker | 3 | 21 | 6 | 5 | 6 | 8 | 27 | 1 | 0 | 35 | 112 |
| Σ | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 135 | 1350 |
| Class-wise accuracy of estimation in % | | | | | | | | | | | |
| | 19.26 | 60.00 | 24.44 | 33.33 | 80.00 | 41.48 | 37.04 | 57.04 | 37.78 | 25.93 | |

regions, the shape and size of which are data driven. In principle every adaptive metric method could be extended following our suggestion, but we consciously choose LVQ because of its easily interpretable results and the lower computational costs in comparison to other approaches. Similarly to Gabor filters any other family of 2D filters commonly used to describe gray scale image information could be adapted and applied to color image analysis with this algorithm. Initializing with a filter bank of differences of Gaussians for color edge detection is a possible example. Furthermore, depending on the task at hand it might be desirable that two patches in which the same texture occurs on different positions should not be interpreted as similar. In this case another similarity measure should be used:

$\|\mathbf{r}(\mathbf{x}^i) - \mathbf{r}(\mathbf{w}^L)\|^2$, which is not based on the difference of magnitudes. This might be of advantage for example in the recognition of objects such as traffic signs, where a corner or an edge might have different interpretations dependent on their position in the image. Combinations of CIA-LVQ with keypoint detectors to avoid the drawing of patches from random positions within an image can also be easily implemented and can be beneficial especially for tasks that are related to object recognition. A completely unbiased, regarding the nature of the filters, variant of CIA-LVQ where the adaptive kernels are randomly initialized is also of particular interest mostly in cases where there is no prior knowledge for the nature of the data (i.e. medical imaging).

Fig. 9 Plots of the optimized prototypes $|\mathbf{w}^L|$ actively used to classify the data in the evaluation set of the KTH-TIPS database. The names consist of the corresponding class name and the index number (1–4) of the prototype

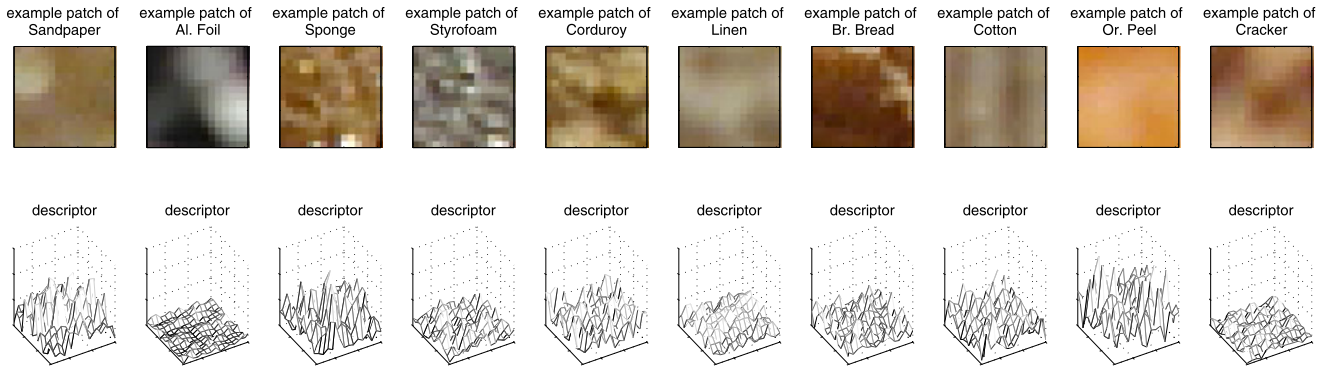
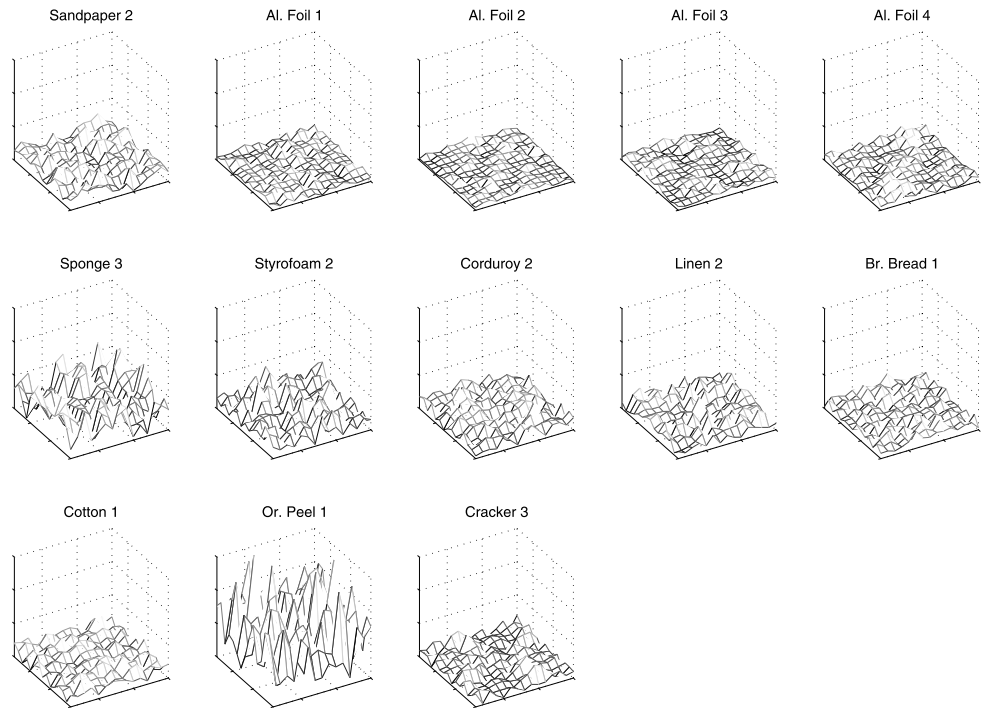


Fig. 10 Plots of the descriptors $|\mathbf{r}_k|$ of some correctly classified image patches from the evaluation set of KTH-TIPS database

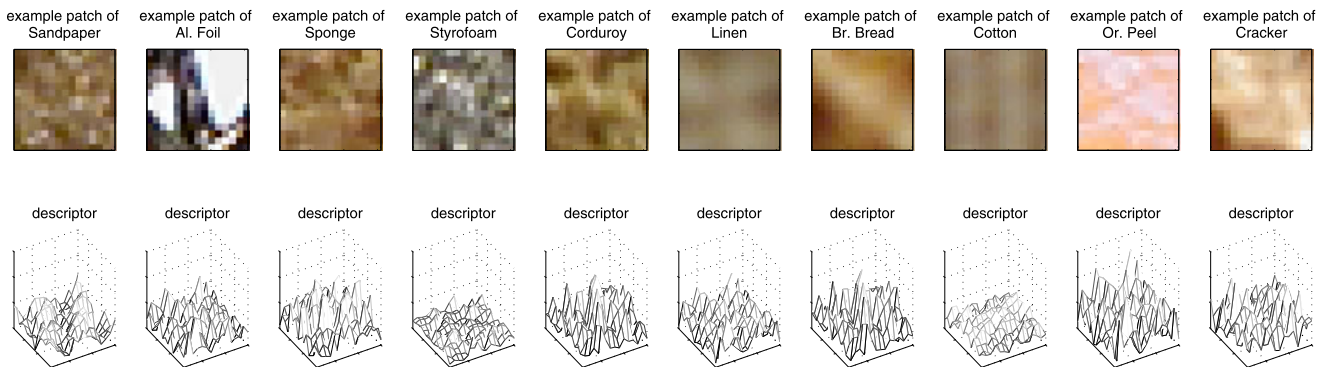
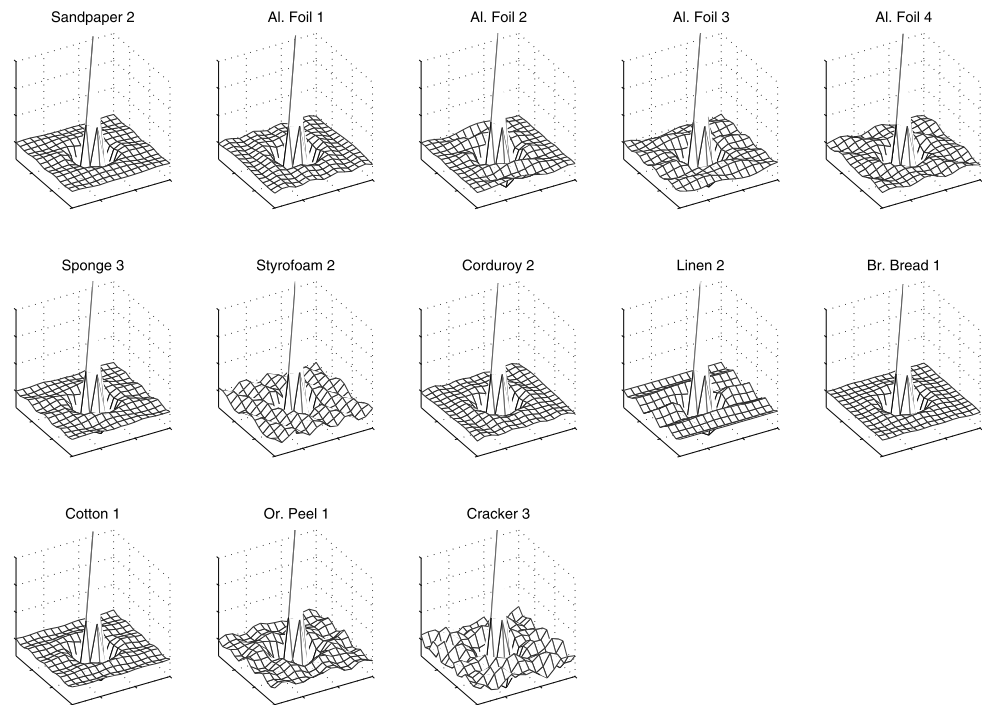


Fig. 11 Plots of the descriptors $|\mathbf{r}_k|$ of some wrongly classified image patches from the evaluation set of KTH-TIPS database

Fig. 12 Plots of the final form of filter kernels actively used to classify the evaluation set of the VisTex database. The filter kernels have been locally adapted during training. The names consist of the corresponding class name and the index number (1–4) of the kernel



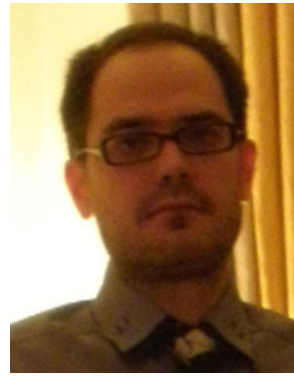
CIA-LVQ formulates a novel general principle: based on a differentiable convolution and an adaptive filter bank, the algorithm optimizes the classification. Contrary to standard approaches which are either based on a single channel representation of the images through a fixed transformation or empirical observations for combining color and textural information, the proposed technique offers the alternative of data driven learning of suitable, parameterized image descriptors. The ability of automatically weighing different color channels and different filters in localized neighborhoods, according to their importance for the classification task, is the most significant factor which qualifies our approach.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

- Bunte, K., Biehl, M., Jonkman, M.F., Petkov, N.: Learning effective color features for content based image retrieval in dermatology. *Pattern Recognit.* **44**(9), 1892–1902 (2011)
- Bunte, K., Hammer, B., Schneider, P., Biehl, M.: Nonlinear discriminative data visualization. In: Verleysen, M. (ed.) *Proc. of European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, pp. 65–70 (2009)
- Database VisTex of Color Textures from MIT. <http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html>
- Daugman, J.G.: Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A* **2**(7), 1160–1169 (1985). doi:[10.1364/JOSAA.2.001160](https://doi.org/10.1364/JOSAA.2.001160)
- Drimbarean, A., Whelan, P.F.: Experiments in colour texture analysis. *Pattern Recognit. Lett.* **22**(10), 1161–1167 (2001). doi:[10.1016/S0167-8655\(01\)00058-7](https://doi.org/10.1016/S0167-8655(01)00058-7)
- Fogel, I., Sagi, D.: Gabor filters as texture discriminator. *Biol. Cybern.* **61**(2), 103–113 (1989). doi:[10.1007/BF00204594](https://doi.org/10.1007/BF00204594)
- Fritz, M., Hayman, E., Caputo, B., Eklundh, J.O.: The KTH-TIPS database. www.nada.kth.se/cvap/databases/kth-tips
- Geusebroek, J.M., van den Boomgaard, R., Smeulders, A.W.M., Geerts, H.: Color invariance. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(12), 1338–1350 (2001)
- Grigorescu, S., Petkov, N., Kruizinga, P.: Comparison of texture features based on Gabor filters. *IEEE Trans. Image Process.* **11**(10), 1160–1167 (2002). doi:[10.1109/TIP.2002.804262](https://doi.org/10.1109/TIP.2002.804262)
- Han, J., Ma, K.K.: Rotation-invariant and scale-invariant Gabor features for texture image retrieval. *Image Vis. Comput.* **25**(9), 1474–1481 (2007). doi:[10.1016/j.imavis.2006.12.015](https://doi.org/10.1016/j.imavis.2006.12.015)
- Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **3**(6), 610–621 (1973). doi:[10.1109/TSMC.1973.4309314](https://doi.org/10.1109/TSMC.1973.4309314)
- Hauta-Kasari, M., Parkkinen, J., Jaaskelainen, T., Lenz, R.: Multi-spectral texture segmentation based on the spectral cooccurrence matrix. *PAA Pattern Anal. Appl.* **2**, 275–284 (1999)
- Hering, E.: *Outlines of a Theory of the Light Sense*, 1st edn. Harvard University Press, Cambridge (1964)
- Hoang, M.A., Geusebroek, J.M., Smeulders, A.W.: Color texture measurement and segmentation. *Signal Process.* **85**(2), 265–275 (2005). doi:[10.1016/j.sigpro.2004.10.009](https://doi.org/10.1016/j.sigpro.2004.10.009). SI on Content Based Image and Video Retrieval
- Jain, A., Healey, G.: A multiscale representation including opponent color features for texture recognition. *IEEE Trans. Image Process.* **7**(1), 124–128 (1998). doi:[10.1109/83.650858](https://doi.org/10.1109/83.650858)

16. Jain, A.K., Farrokhnia, F.: Unsupervised texture segmentation using Gabor filters. *Pattern Recognit.* **24**(12), 1167–1186 (1991). doi:[10.1016/0031-3203\(91\)90143-S](https://doi.org/10.1016/0031-3203(91)90143-S)
17. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer, Berlin (2001)
18. Kruijzinga, P., Petkov, N.: A computational model of periodic-pattern-selective cells. In: *From Natural to Artificial Neural Computation. Lecture Notes in Computer Science*, vol. 930, pp. 90–99. Springer, Berlin (1995)
19. Mäenpää, T., Pietikäinen, M.: Classification with color and texture: jointly or separately? *Pattern Recognit.* **37**(8), 1629–1640 (2004). doi:[10.1016/j.patcog.2003.11.011](https://doi.org/10.1016/j.patcog.2003.11.011)
20. Mäenpää, T., Pietikäinen, M., Viertola, J.: Separating color and pattern information for color texture discrimination. In: *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*. ICPR '02, vol. 1, p. 10,668. IEEE Comput. Soc., Los Alamitos (2002)
21. Manjunath, B., Ma, W.: Texture features for browsing and retrieval of image data. *IEEE Trans. Pattern Anal. Mach. Intell.* **18**(8), 837–842 (1996). doi:[10.1109/34.531803](https://doi.org/10.1109/34.531803)
22. Messer, K., Kittler, J.: A region-based image database system using colour and texture. *Pattern Recognit. Lett.* **20**(11–13), 1323–1330 (1999). doi:[10.1016/S0167-8655\(99\)00101-4](https://doi.org/10.1016/S0167-8655(99)00101-4)
23. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. *Pattern Recognit.* **29**(1), 51–59 (1996)
24. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **24**, 971–987 (2002)
25. Palm, C.: Color texture classification by integrative co-occurrence matrices. *Pattern Recognit.* **37**(5), 965–976 (2004). doi:[10.1016/j.patcog.2003.09.010](https://doi.org/10.1016/j.patcog.2003.09.010)
26. Papari, G., Bunte, K., Biehl, M.: Waypoint averaging and step size control in learning by gradient descent. Technical Report MLR-2011-06, Leipzig University (2011). http://www.techfak.uni-bielefeld.de/~fschleif/mlr/mlr_06_2011.pdf
27. Paschos, G.: Fast color texture recognition using chromaticity moments. *Pattern Recognit. Lett.* **21**(9), 837–841 (2000). doi:[10.1016/S0167-8655\(00\)00043-X](https://doi.org/10.1016/S0167-8655(00)00043-X)
28. Pietikäinen, M., Mäenpää, T., Viertola, J.: Color texture classification with color histograms and local binary patterns. In: *IWTAS*, pp. 109–112 (2002)
29. Pietikäinen, M., Ojala, T., Xu, Z.: Rotation-invariant texture classification using feature distributions. *Pattern Recognit.* **33**(1), 43–52 (2000). doi:[10.1016/S0031-3203\(99\)00032-1](https://doi.org/10.1016/S0031-3203(99)00032-1)
30. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (eds.) *Advances in Neural Information Processing Systems*, vol. 8, pp. 423–429. MIT Press, Cambridge (1996)
31. Schneider, P., Biehl, M., Hammer, B.: Adaptive relevance matrices in learning vector quantization. *Neural Comput.* **21**(12), 3532–3561 (2009)
32. Turner, M.R.: Texture discrimination by Gabor functions. *Biol. Cybern.* **55**, 71–82 (1986). doi:[10.1007/BF00341922](https://doi.org/10.1007/BF00341922)
33. Wang, J.W., Chen, C.H., Chien, W.M., Tsai, C.M.: Texture classification using non-separable two-dimensional wavelets. *Pattern Recognit. Lett.* **19**, 1225–1234 (1998). doi:[10.1016/S0167-8655\(98\)00105-6](https://doi.org/10.1016/S0167-8655(98)00105-6)
34. Wang, L., Liu, J.: Texture classification using multiresolution Markov random field models. *Pattern Recognit. Lett.* **20**(2), 171–182 (1999). doi:[10.1016/S0167-8655\(98\)00129-9](https://doi.org/10.1016/S0167-8655(98)00129-9)



Ioannis Giotis received the Diploma (2006) in Informatics and Telecommunications from the University of Athens, Greece and the M.Sc. degree with distinction (2008) in Technology Education and Digital Systems from the University of Piraeus, Greece. He has been an associate of the Institute of Informatics and Telecommunications of the National Center for Scientific Research, Athens, Greece. Currently, he is a PhD candidate at the Johann Bernoulli Institute for Mathematics and Computer Science of the University of Groningen, The Netherlands. His current research is in the field of pattern recognition focusing on the extraction and learning of visual patterns and their applications in the health care domain. Further information can be obtained from <http://www.cs.rug.nl/~jannis/>



Kerstin Bunte graduated at the Faculty of Technology at the University of Bielefeld, Germany, and joined the Institute of Mathematics and Computing Science of the University of Groningen, The Netherlands, in September 2007. She received a Ph.D. in Computer Science in December 2011. Currently she is working at the Bielefeld University at the Cognitive Interaction Technology Center of Excellence (CITEC). Her recent work has focused on Machine Learning techniques, especially Learning Vector Quantisation and their usability in the field of image processing, supervised dimension reduction and visualization. Further information can be obtained from <http://www.cs.rug.nl/~kbunte/>



Nicolai Petkov received the Dr.sc.techn. degree in Computer Engineering (Informationstechnik) from Dresden University of Technology, Dresden, Germany. He is professor of computer science and head of the Intelligent Systems group of the Johann Bernoulli Institute of Mathematics and Computer Science of the University of Groningen, the Netherlands. He is the author of two monographs and coauthor of another book on parallel computing, holds four patents and has authored over 100 scientific papers. His current research is in image processing, computer vision and pattern recognition, and includes computer simulations of the visual system of the brain, brain-inspired computing, computer applications in health care and life sciences and creating computer programs for artistic expression. Prof. dr. Petkov is a member of the editorial boards of several journals.



Michael Biehl received a Ph.D. in Theoretical Physics from the University of Giessen, Germany, in 1992 and the *venia legendi* in Theoretical Physics from the University of Würzburg, Germany, in 1996. He is currently Professor in Computing Science at the University of Groningen, The Netherlands. His main research interest is in the theory, modelling and application of Machine Learning techniques. He is furthermore active in the modelling and simulation of complex physical systems. He has co-authored

more than 100 publications in international journals and conferences; preprint versions and further information can be obtained from <http://www.cs.rug.nl/~biehl/>