Local Higher-Order Statistics (LHS) describing images with statistics of local non-binarized pixel patterns

Gaurav Sharma, Frédéric Jurie

Local Higher-Order Statistics (LHS) describing images with statistics of local non-binarized pixel patterns

Gaurav Sharma[a,b,1], Frédéric Jurie[a]

[a] *GREYC CNRS UMR 6072, Université de Caen Basse-Normandie, France*
[b] *Max Planck Institute for Informatics, Germany*

Abstract

We propose a new image representation for texture categorization and facial analysis, relying on the use of higher-order local differential statistics as features. It has been recently shown that small local pixel pattern distributions can be highly discriminative while being extremely efficient to compute, which is in contrast to the models based on the global structure of images. Motivated by such works, we propose to use higher-order statistics of local non-binarized pixel patterns for the image description. The proposed model does not require either (i) user specified quantization of the space (of pixel patterns) or (ii) any heuristics for discarding low occupancy volumes of the space. We propose to use a data driven soft quantization of the space, with parametric mixture models, combined with higher-order statistics, based on Fisher scores. We demonstrate that this leads to a more expressive representation which, when combined with discriminatively learned classifiers and metrics, achieves state-of-the-art performance on challenging texture and facial analysis datasets, in low complexity setup. Further, it is complementary to higher complexity features and when combined with them improves performance.

Keywords: local features, texture categorization, face verification, image classification.

1. Introduction

Categorization of textures and analysis of faces under multiple and difficult sources of variations like illumination, scale, pose, expression and appearance etc. are challenging problems in computer vision with many important applications. Texture recognition is beneficial for applications such as mobile robot navigation or biomedical image processing. It is also related to facial analysis e.g. facial expression categorization and face verification (two faces are of same person or not), as the models developed for textures are generally found to be competitive for face analysis. Analysis of faces, similarly, has important applications especially in human computer interaction and in security and surveillance scenarios. This paper proposes a new model for obtaining a powerful and highly efficient representation for textures and faces, with such applications in mind.

Initial success on texture recognition was achieved by the use of filter banks [4, 5, 6, 7, 8], where the distributions of the filter response coefficients were used for discrimination. The focus was on evaluating appropriate filters, selective for edge orientation and spatial-frequencies of variations, and better capturing the distributions of such filter responses. However, later works e.g. by Ojala et al. [9] and Varma and Zisserman [10], showed that it is possible to discriminate between textures using pixel values directly (with pixel neighborhoods as small as 3×3 pixels), discounting the necessity of filter banks. It was demonstrated that despite the global structure of the textures, very good discrimination could be achieved by exploiting the distributions of such small pixel neighborhoods. More recently, exploiting such small pixel neighborhoods or *micro-structures* in textures by representing images with distributions of local descriptors has gained much attention and has led to state-of-the-art performances for systems with low complexity, e.g. Local Binary Patterns (LBP) [1, 2], Local Ternary Patterns (LTP) [11] and Weber Local Descriptor (WLD) [12]. Most of such local pixel neighborhood based descriptors were shown to be highly effective for facial analysis [2, 11] as well. However these methods suffer from important limitations–the use of fixed hard quantization of the feature space (the space of small pixel patterns) and the use of heuristics to prune uninteresting regions in the feature space. In addition, they use histograms to represent the feature distributions. Histograms, or count statistics, are zeroth order statistics of distributions and thus give a quite restrictive representation.

In contrast, we propose a model that represents images with higher-order statistics of small local pixel neighborhoods. Fig. 1 shows an illustration of this representation.

Email addresses: grvsharma@gmail.com (Gaurav Sharma), frederic.jurie@unicaen.fr (Frédéric Jurie)

[1] GS is currently with Max Planck Institute for Informatics. The majority of the work was done when GS was with the Université de Caen Basse-Normandie

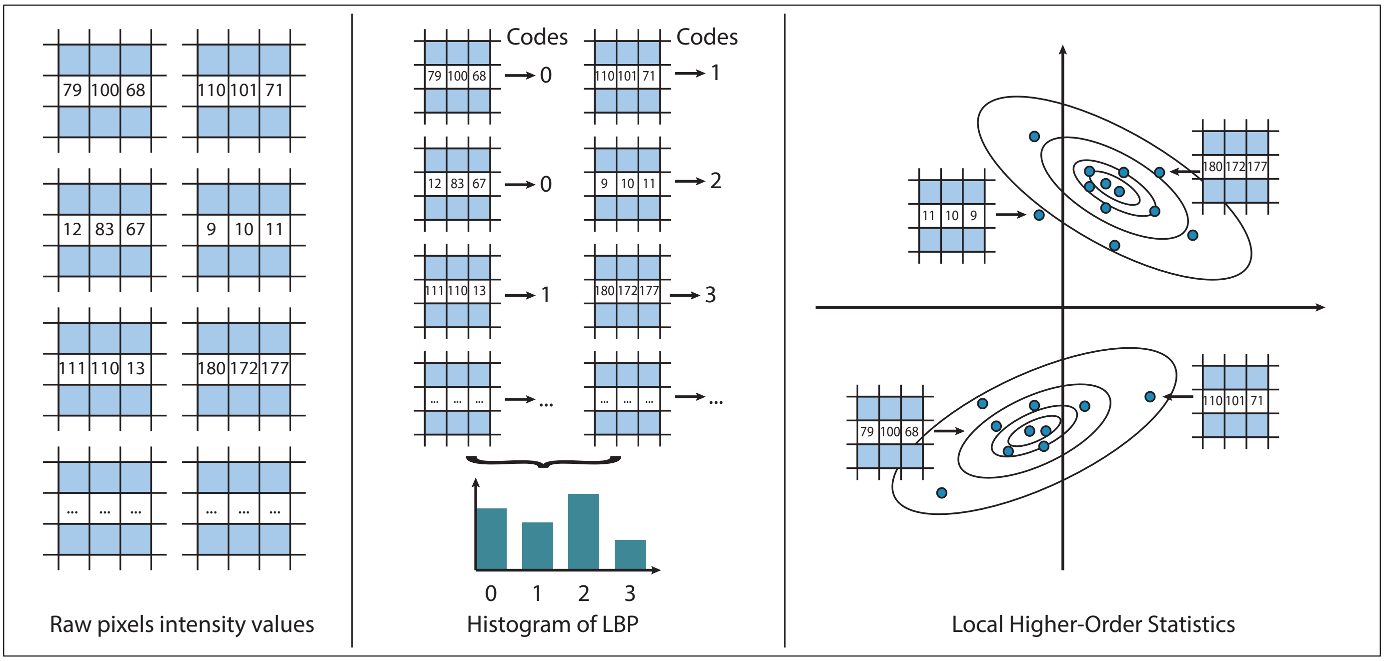

Figure 1: Illustration of the proposed Local Higher-order Statistics (LHS) representation. The left-hand side of the figure represents a collection of pixel-centered raw pixel intensity values (image patches). For making the figure simple, we consider only horizontal 2-neighborhood (in practice we use a 3x3 neighborhood). The middle of the figure shows how these patches can be turned into LBP codes [1, 2] – 4 different codes in this case – and then represented as an histogram of LBP. The proposed representation, illustrated on the right-hand side of the figure, is much richer, as the distribution of the local patches is represented by a Gaussian Mixture model, encoded as Fisher scores [3].

We obtain a data driven soft partition of the feature space using parametric mixture models, to represent the distribution of the vectors, with the parameters learnt from the training data. Hence, in the proposed method, the coding of vectors is intrinsically adapted to the data and the computations involved remain very simple despite the strengths. This helps us avoid the above mentioned limitations of the previous methods – (i) instead of a fixed quantization, we learn a data driven, and hence, adaptive quantization using Gaussian mixture models (GMM), (ii) quantizing using GMM also avoids any heuristic pruning as any low occupancy region in the feature space will be automatically ignored by the GMM learning and (iii) learning GMM allows us to use Fisher vectors [3] which are higher-order statistics of the feature distribution. We discuss in more detail on this in the following sections. A preliminary version of this work appeared in Sharma et al. [13].

We validate the proposed representation by extensive experiments on four challenging datasets: (i) Brodatz 32 texture dataset [14, 15], (ii) KTH TIPS 2a materials dataset [16], (iii) Japanese Female Facial Expressions (JAFFE) dataset [17], and (iv) Labeled Faces in the Wild (LFW) dataset [18]. Two dataset, Brodatz-32 and JAFFE, are relatively easier with limited variations while the other two, KTH TIPS 2a and LFW, are more challenging with realistic high levels of variation in illumination, pose, expressions etc. We show that using higher-order statistics gives a more expressive description and lead to state-of-the-art performance in low complexity settings, for the above datasets. Further, with the challenging LFW dataset as the experimental testbed, we also show that the proposed representation is complementary to the recent high complexity state-of-the-art representations. However, in case of challenging variations, like in LFW, unsupervised approach is not sufficient and hence we show that when used with supervised metric learning the performance of the proposed representation improves substantially. When combined with higher complexity methods, the proposed representation achieves the state-of-the-art performance on the challenging LFW dataset in the supervised protocol, when no external labeled data is used.

2. Related works

Texture analysis was initially addressed using filter banks and the statistical distributions of their responses e.g. [4, 5, 6, 7, 8]. Most of the initial works proposed appropriate directionally and frequency-adapted multiscale filter banks and/or methods to better capture the statistical distributions of their responses. Later, Ojala et al. [9] and, more recently, Varma and Zisserman [10] showed that statistics of small pixel neighborhoods, as small as 3×3 pixels, are capable of achieving high discrimination. This was in contrast to first convolving the local patches with filter banks and then taking their responses. The success of using raw pixel patches without any processing discounted the use of filter banks for texture recognition. Since then many methods working directly with local pixel neighborhoods have been used successfully in texture and face analysis e.g. Local Binary Patterns (LBP) [1, 2], Local Ternary Patterns (LTP) [11] and Weber Law Descriptor (WLD) [12].

Local pixel pattern operators, such as Local Binary Patterns (LBP) by Ojala et al. [9], have been very successful for image description. LBP based image representation aims to capture the joint distribution of pixel intensities in a local neighborhood as small as 3×3 pixels. LBP makes two approximations, (i) it takes the differences between the center pixel and its eight neighbors and (ii) then considers just the signs of the differences. The first approximation lends invariance to gray-scale shifts and the second to intensity scaling. As an extension to LBP, Local Ternary Patterns (LTP) were introduced by Tan and Triggs [11] to add resistance to noise. LTP adds an additional parameter t, which defines a tolerance for similarity between different gray intensities, allowing for robustness to noise. Doing so lends an important strength: LTPs are capable of encoding pixel similarity information modulo noise using the simple rule that any two pixels within $\pm t$ intensity of each other are considered similar. This is accompanied by a clever split coding scheme to control the size of the descriptor. However, LTP (and LBP) coding is still limited due to its hard and fixed quantization. In addition, both LBP and LTP representations usually use the so-called *uniform* patterns: patterns with at most one 0-1 and at most one 1-0 transition, when seen as circular bit strings. It was empirically observed that that uniform patterns account for nearly 90% of all observed pixel patterns in textures, and hence ignoring the non-uniform patterns leads to large savings in space at negligible loss of accuracy. Although uniform patterns are beneficial in practice, their use is still a heuristic for discarding low occupancy volumes in feature space. We will discuss this in more detail in Sec. 3.3

Owing to the success of the *texton* based texture classification method, e.g. Leung and Malik [6], and the recent success of *bag of words* representation for image retrieval (by Sivic and Zisserman [19]) and classification (by Csurka et al. [20]) many of the recent methods for texture and face analysis, e.g. [10, 21, 22, 23, 24, 25, 26, 27, 28], use histogram based representations. They first compute a dictionary or codebook of prototypical vectors, so-called textons or visual words, by clustering large number of randomly sampled vectors from the training data. The images are then represented as histograms over the learnt codebook texton assignments. The local vectors are derived in multiple ways, incorporating different invariances like rotation, view point etc. E.g. [22, 23] generate an image specific texton representation from rotation and scale invariant descriptors and compare them using Earth Movers distance, whereas [10, 9, 21, 24] use a dictionary learned over the complete dataset to represent each image as histogram over this dictionary.

In a more recent line of work, Cimpoi et al. [29] show that traditional image classification methods when applied to the more challenging textures in the wild scenario give good performances. They use classic local features such as the Scale Invariant Feature Transform (SIFT) [30] with different encoding methods, particularly Fisher scores [3], similar to those employed in the present work. They also evaluate deep learning [31] based representation and demonstrate their usefulness for the task. We note that while these method give good performances, they are of much higher complexities than the proposed method. The proposed method is also complementary to such methods as we will show empirically later.

We can thus draw a few conclusions from the above mentioned previous works. Modeling distributions of small pixel neighborhoods (as small as 3×3 pixels) can be quite effective for image representation [9, 10, 11]. However, using coarse approximations (we discuss more on this in Sec. 3.3), as done by most of the previous related approaches, limits their potential. Finally, the previous methods use low-order statistics, generally zeroth order counts i.e. histograms. This is also limiting as using high-order statistics can give a more accurate and expressive representation. The main contribution of this paper is motivated by these observations; we describe small neighborhoods with their higher-order statistics, without coarse approximations, and show with extensive experimental results that this leads to a more expressive representation which performs better on challenging benchmark datasets.

In more recent works on facial analysis, deep learning based methods for face recognition/verification [32, 33, 34, 35, 36, 37, 38, 39, 40, 41] have gained much success. Most of the deep learning based works aim to leverage large amount of data along with the impressive model capacity of deep networks. Taigman et al. [38] showed that learning a deep convolutional neural network, for predicting thousands of identities using millions of training images, and then using the output of the penultimate layer of trained network as features for faces results in very good face verification performance. Alternatively, Huang et al. [33] and Schroff et al. [35] proposed deep architectures to perform metric learning directly. Kan et al. proposed to handle high variations to pose [40] while Schroff et al. [35] propose to use the obtained embeddings to cluster faces based on identities.

The approach of Martinez [42] is also related to the proposed approach, but the two are complementary. Martinez [42] proposed to divide face images into small number of (typically six) local regions and then learn a Gaussian mixture model on PCA compressed pixel representation of the local face regions. Further, for expression invariant recognition, Martinez [42] proposed to learn weights on the local regions of the face corresponding to how important (or, in some sense, invariant) the different regions are, for recognition of expression variant faces. The proposed method learns the description of an image by the statistics of very local (3×3 pixel) neighborhoods. Hence the relatively larger in size (six) local regions in Martinez's approach can be represented by LHS vectors, instead of vectorized raw pixels compressed with PCA. LHS vectors could be seen as one extreme (highly local) representation of images, with the PCA based fully global representation at the other extreme. Martinez's approach can then be seen as striking a balance between the two.

3. The Local Higher-order Statistics (LHS) model

We now describe the main contribution of the paper– Local Higher-order Statistics (LHS) model. LHS intends to represent images by accurately describing the distribution of local pixel neighborhoods using higher-order statistics. We start with small pixel neighborhoods of 3×3 pixels and model the statistics of their local differential vectors.

3.1. Local differential vectors.

Consider all possible 3 × 3 pixel neighborhoods in an image, i.e.

$$\mathbf{v}^n = (v_c, v_1, \ldots, v_8) \tag{1}$$

where v_c is the intensity of the center pixel and the rest are those of its 8-neighbors. We are interested in exploiting the distribution $p(\mathbf{v}^n|I)$ of the these vectors to represent the image. Following LBP [1], to obtain invariance to monotonic changes in gray levels, we subtract the value of the center pixel from the rest and obtain the local differential vectors i.e.

$$\mathbf{v} = (v_1 - v_c, \ldots, v_8 - v_c). \tag{2}$$

We approximate the distribution of the local pixel patterns with the distribution of the corresponding differential vectors i.e.

$$p(\mathbf{v}^n|I) \approx p(\mathbf{v}|I). \tag{3}$$

3.2. Higher order statistics.

As the key contribution, we propose to characterize the images using the higher-order statistics of the differential vectors. We avoid a hard and/or predefined quantization, as used in LBP/LTP, and use parametric Gaussian mixture model (GMM) to obtain a probabilistic partitioning of the differential space (i.e. the space of all differential vectors). Defining such soft quantization with mixture model can be equivalently seen as a generative model on the differential vectors. It allows us to use a characterization method which exploits higher-order statistics i.e. *Fisher score* method proposed by Jaakkola and Haussler [3]. Fisher scores enables the use of generative modeling with discriminative classifiers. The key idea is to obtain a fixed length representation of set of vectors, of arbitrary cardinality, by representing each of the vectors with gradients wrt. the generative model and averaging their representations (with an iid assumption). More precisely, given a parametric generative model, a vector $\mathbf{v}$ is characterized by the gradient of the log likelihood, computed at $\mathbf{v}$, with respect to the parameters of the model. The Fisher score, for an observed vector $\mathbf{v}$ wrt. a distribution $p(\mathbf{v}|\boldsymbol{\lambda})$, is given as,

$$g(\boldsymbol{\lambda}, \mathbf{v}) = \nabla_{\boldsymbol{\lambda}} \log p(\mathbf{v}|\boldsymbol{\lambda}), \tag{4}$$

where $\boldsymbol{\lambda}$ is the parameter vector. The Fisher score vector, thus, has the same dimensions as the parameter vector $\boldsymbol{\lambda}$. In the case of a mixture of Gaussian distribution i.e. when

$$p(\mathbf{v}|\boldsymbol{\lambda}) = \sum_{k=1}^{N_k} \alpha_k \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{5}$$

$$\mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^d |\boldsymbol{\Sigma}_k|}} e^{-\frac{1}{2}(\mathbf{v}-\boldsymbol{\mu}_k)\boldsymbol{\Sigma}_k^{-1}(\mathbf{v}-\boldsymbol{\mu}_k)}, \tag{6}$$

the Fisher scores can be computed using the following partial derivatives

$$\frac{\partial \log p(\mathbf{v}|\boldsymbol{\lambda})}{\partial \boldsymbol{\mu}_k} = \gamma_k \boldsymbol{\Sigma}_k^{-1}(\mathbf{v} - \boldsymbol{\mu}_k) \tag{7a}$$

$$\frac{\partial \log p(\mathbf{v}|\boldsymbol{\lambda})}{\partial \boldsymbol{\Sigma}_k^{-1}} = \frac{\gamma_k}{2}\left(\boldsymbol{\Sigma}_k - (\mathbf{v} - \boldsymbol{\mu}_k)^2\right) \tag{7b}$$

$$\text{where,} \quad \gamma_k = \frac{\alpha_k p(\mathbf{v}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_k \alpha_k p(\mathbf{v}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}, \tag{7c}$$

with the square of a vector being done element-wise. We have assumed diagonal $\boldsymbol{\Sigma}$, to decrease the number of parameters to be learnt. This amounts to assuming statistical independence between the variables. Thus, coding vectors using Eq. 7 codes the higher-order, i.e. based on the first and second power of $\mathbf{v}$, statistics of the local differential vectors. After obtaining the Fisher scores of differential vectors corresponding to every pixel neighborhood in the image, we compute the image representation as the average of the Fisher scores over all of them. Here we make an implicit assumption that the vectors were generated iid from the distribution. This way any image of arbitrary size or equivalently with arbitrary number of vectors is represented as a vector of length equal to the number of parameters.

We then perform the following normalizations; first, we normalize each dimension of the image vector to zero mean and unit variance. To perform the normalization we use training vectors and compute multiplicative and additive constants to perform whitening per dimension [43]. Second, we perform power normalization on the image vector $\mathbf{x}$,

$$(x_1, \ldots, x_d) \leftarrow (\text{sign}(x_1)\sqrt{|x_1|}, \ldots, \text{sign}(x_d)\sqrt{|x_d|}), \tag{8}$$

and finally we do ℓ_2 normalization of $\mathbf{x}$,

$$(x_1, \ldots, x_d) \leftarrow \left(\frac{x_1}{\sqrt{\sum x_i^2}}, \ldots, \frac{x_d}{\sqrt{\sum x_i^2}}\right). \tag{9}$$

Perronnin et al. [44] motivate the power normalization for obtaining a *de-sparsification* effect, which makes the use of ℓ_2 distance (and hence the corresponding linear support vector machine) more appropriate. Similar power normalization has also been shown to be an *explicit feature map* by Vedaldi and Zisserman [45] i.e. a mapping which transforms the vectors to a space where the dot product of the transformed vectors corresponds to the Bhattacharyya

Algorithm 1 Computing Local Higher-order Statistics

```
Randomly sample 3×3 pixels differential vectors $\{\mathbf{v} \in I | I \in \mathcal{I}_{train}\}$
Learn the GMM parameters $\{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k | k = 1 \dots K\}$ with EM algorithm on $\{\mathbf{v}\}$
Compute the higher-order statistics, i.e. Fisher scores, for $\{\mathbf{v}\}$ using Eq. (7)
Compute means $C^i_\mu$ and variances $C^i_\Sigma$ for each coordinate $i \in \{1, \dots, d_0\}$
for all images $\{I\}$ do
    Compute all differential vectors $\mathbf{v} \in I$
    Compute the Fisher scores for all features $\{\mathbf{v}\}$ using Eq. (7)
    Compute the image representation $\mathbf{x}$ as the average score over all features
    Normalize each coordinate $i$ as $x^i \leftarrow (x^i - C^i_\mu)/C^i_\Sigma$
    Apply normalizations, Eq. (8) and (9)
end for
```

kernel between the original vectors. The whole algorithm, which is remarkably simple, is summarized in Alg. 1.

Finally, we use the vectors obtained as the representation of the images and employ either discriminative linear support vector machine (SVM) for supervised classification tasks or discriminatively learnt Mahalanobis like metrics (detailed below in Sec. 4) for supervised pair matching, i.e. verification, task.

3.3. Relation to LBP/LTP.

We now discuss how LHS can be considered as a generalization of local pattern features. Consider the Local Binary Patterns (LBP) of Ojala et al. [9]–every pixel is coded as a binary vector of 8 bits corresponding to its 8 immediate neighbors. Each bit of LBP indicates whether the corresponding neighboring pixels is of greater intensity than the current pixel or not. We can thus derive LBP [9] by thresholding each coordinate of our differential vectors at zero. Hence the LBP space can be seen as a discretization of the differential space into two bins per coordinate, i.e. into the 2^8 hyperoctants of the 8-dimensional space of local differential vectors. Similarly, we can discretize the differential space into more number of bins, with three bins per coordinate i.e. $(-\infty, -t), [-t, t], (t, -\infty)$ we arrive at the local ternary patterns [11] and so on. The use of *uniform patterns* (patterns with exactly one 0-1 and one 1-0 transition), in both LBP/LTP, can be seen as an empirically derived heuristic for ignoring volumes, in differential space, which have low occupancies, e.g. more than 75% of the hyperoctants for LBP[2]. Thus, the local binary/ternary patterns are obtained with (i) a hard and hand set quantization of space and (ii) a rejection heuristic derived from empirical observation. While for LHS such quantization of space is learnt from data using parametric mixture models, which automatically adapts itself locally according to the occupancy levels of the space. Hence, in our case the quantization and rejection is data driven and more general. Moreover, in LBP/LTP the final representation is based on zeroth order statistics, i.e. counts/histograms, while using a data driven soft quantization allows us to exploit higher-order statistics, as detailed above, for a more expressive image description.

Algorithm 2 SGD for distance learning

```
Given: Training set ($\mathcal{T}$), bias ($b$), margin ($m$), learning rate ($r$)
Initialize: $L, V \leftarrow$ Whitened PCA of randomly selected training faces $\{\mathbf{x}\}$
for all $i = 1, \dots,$niters do
    Randomly sample a face pair $(\mathbf{x}_i, \mathbf{x}_j, y_{ij})$ from $\mathcal{T}$
    Compute $D^2_J(\mathbf{x}_i, \mathbf{x}_j)$ using Eq. 10
    if $y_{ij}(b - D^2_J(\mathbf{x}_i, \mathbf{x}_j)) < m$ then
        $L \leftarrow L - r y_{ij} L(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top$
        $V \leftarrow V + r y_{ij} V \mathbf{x}_i \mathbf{x}_j^\top$
    end if
end for
```

4. Discriminative metric learning

Recently it has been shown that popular features can be compressed by orders of magnitude by learning low dimensional projections with a discriminative objective function for the task of pair matching i.e. verification. Such supervised learning also enhances the discrimination capability of the features upon projection. In the experimental section, we show the efficacy of the proposed Local Higher-order Statistics (LHS) features when used with discriminative learning for the challenging task of face verification. In this section, we give the details of the discriminative metric learning method we use to learn such projection.

Metric learning has recently been a popular topic of research in the machine learning community. While an exhaustive review of different metric learning methods is out of scope of the paper, we encourage the interested reader to see an excellent review by Bellet et al. [46]. More closely related to the present work, metric learning has been successfully applied to the task of face verification, i.e. to predict if two images are of the same person or not. This is different from face recognition, as the faces may be of person(s) never seen before. The discriminative objectives used in such methods are based usually on margin maximizing or probabilistic principles [47, 48, 49, 50]. Inspired by such works we now present the method we use to learn a metric using the proposed LHS face representation.

We are interested in learning a 'distance' function, for comparing two faces $\mathbf{x}_i$ and $\mathbf{x}_j$, parameterized by two matrices L and V. Our function $D_J(\cdot)$ is a combination of two terms, first term $D_L(\cdot)$ is the Euclidean distance in

[2] Out of the total 256 bins for all possible $8d$ binary patterns, 58 bins for uniform patterns and one bin for all the rest of the patterns, are usually used in LBP

the low dimensional space corresponding to the row space of L and the second $D_V(\cdot)$, is the dot product similarity in another low dimensional space corresponding to the row space of V i.e.

$$D_J^2(\mathbf{x}_i, \mathbf{x}_j) = D_L^2(\mathbf{x}_i, \mathbf{x}_j) - D_V^2(\mathbf{x}_i, \mathbf{x}_j) \qquad (10)$$

$$D_L^2(\mathbf{x}_i, \mathbf{x}_j) = \|L\mathbf{x}_i - L\mathbf{x}_j\|^2 = (\mathbf{x}_i - \mathbf{x}_j)^\top L^\top L(\mathbf{x}_i - \mathbf{x}_j) \qquad (11)$$

$$D_V^2(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top V^\top V \mathbf{x}_j, \qquad (12)$$

where we use the subscript 'J' to signify joint Euclidean distance and dot product similarity based distance. Both the matrices L and V map the original d_0 dimensional LHS features to $d \ll d_0$[3] dimensional vectors. d is a free parameter and is chosen on a per-task basis (Sec. 5.7).

We learn the projection matrices L and V by minimizing the following loss function,

$$\mathcal{L}(\mathcal{T}; L, V) = \sum_{\mathcal{T}} \max\left(0, m - y_{ij}(b - D_J^2(\mathbf{x}_i, \mathbf{x}_j)\right) \qquad (13)$$

where $\mathcal{T} = \{(\mathbf{x}_i, \mathbf{x}_j, y_{ij})\}$ is the provided training set, with pairs of faces $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^{d_0}$ annotated to be of the same person ($y_{ij} = +1$) or not ($y_{ij} = -1$). Minimization of this margin-maximizing loss encourages the distance, between pairs of faces of same (different) person, to be less (greater) than the bias b by a margin of m.

We learn the parameters, i.e. L and V, with a stochastic gradient descent (SGD) algorithm with easily calculable analytic gradients outlined in Alg. 2.

5. Experimental results

We now report various experimental results which validate the proposed method. We use four challenging publicly available datasets of textures and faces and address the challenging tasks of texture recognition, texture categorization, facial expression categorization and face verification.

In the following, we first discuss implementation details then present the datasets and finally give the experimental results for each dataset. In the first set of experiments (upto Sec. 5.6), as our focus is on rich and expressive representation, we use a standard classification framework based on linear SVM. As linear SVM works directly in the input feature space, any improvement in the performance is directly related to a better encoding of local regions, and thus helps us gauge the quality of our representation vs. the competition, with same setup. In the last part of the experiments (Sec. 5.7), we show results with supervised metric learning, on the LFW dataset, which can also be seen as a discriminatively learned embedding of features.

5.1. Implementation details.

We use only the intensity information of the images and convert color images, if any, to grayscale. We consider two neighborhood sampling strategies (i) rectangular sampling, where the 8 neighboring pixels are used, and (ii) circular sampling, where, like in LBP/LTP [9, 11], we interpolate the diagonal samples to lie on a circle, of radius one, using bilinear interpolation. We randomly sample at most one million features from training images to learn Gaussian mixture model of the vectors, using the EM algorithm initialized with k-means clustering. We keep the number of components as an experimental parameter (Sec. 5.5). We also use these features to compute the normalization constants, by first computing their Fisher score vectors and then computing (per coordinate) mean and variance of those vectors (Alg. 1). We use the average of all the features from the image as the representation for the image. However, for the facial expression dataset we first compute the average vectors for non overlapping cells of 10×10 pixels and concatenate these for all cells to obtain the final image representation. Such gridding helps in capturing spatial information in the image and is standard in face analysis [51, 52]. We crop the 250×250 face images to a ROI of $(66, 96, 186, 226)$, to focus on the face, before feature extraction and do not apply any other pre-processing. Finally, we use linear SVM as the classifier with the cost parameter C set using five fold cross validation on the current training set.

In the supervised setting for face verification, we use the metric learning formulation described above in Sec. 4. We set the bias $b = 1.0$, the margin $m = 0.2$ and rate $r = 0.002$ for all the experiments. During testing a face pair, we horizontally flip the faces and average the distances between the four possible pairs of flipped and non-flipped faces. During training, at each SGD iteration, we randomly select one of the 4 possible flipped/non-flipped pairs for making an update.

We also combine the proposed LHS with our implementation of Fisher Vectors based on dense SIFT features (SIFT-FV) [49, 53, 3]. The implementation is similar to LHS with the local differential vectors being replaced by dense SIFT features. We extract SIFT features, using the `vlfeat` library [54], with a step size of 1 pixel at 5 scales i.e. original image and 2 upsampled and 2 downsampled versions respectively, with a scale difference of $\sqrt{2}$. The SIFT features are compressed to $d_s = 64$ dimension using PCA. We use a vocabulary size of $k = 16$ and use a spatial grid of $N_c = 7 \times 4$, giving a feature of dimension $2 \times k \times d_s \times N_c = 57344$.

5.2. Baselines.

As baselines, we give results with single scale LBP/LTP features generated using the same samplings as our LHS features, in respective experiments. We use histogram representation over bins of uniform patterns and add one bin for all the rest of the patterns. We L1 normalize the histograms and take their square roots and use them with

[3] In general the number of rows of L and V can be different. Here, we keep them the same.

linear SVM. As discussed previously as well, it has been shown that taking square root of histograms transforms them to a space where the dot product corresponds to the non linear Bhattacharyya kernel in the original space [45]. Thus using linear SVM with square root of histograms is equivalent to SVM with non linear Bhattacharyya kernel. Similar square root (i.e. power normalization) was also shown to be useful for Fisher scores [53]. We note here that our baselines are strong baselines.

5.3. Texture categorization

Brodatz – 32 Textures dataset[4] [14, 15] is a standard dataset for texture recognition. It contains 32 texture classes e.g. bark, beach-sand, water, with 16 images per class. Each of the image is used to generate 3 more images by (i) rotating, (ii) scaling and (iii) both rotating and scaling the original image – note that Brodatz-32 [14] dataset is more challenging than original Brodatz dataset and includes both rotation and scale changes. The images are 64×64 pixels histogram normalized grayscale images. We use the standard protocol [12], of randomly splitting the dataset into two halves for training and testing, and report average performance over 10 random splits.

KTH TIPS 2a dataset[5] [16] is a dataset for material categorization. It contains 11 materials e.g. cork, wool, linen, with images of four samples for each material. The samples were photographed at 9 scales, 3 poses and 4 different illumination conditions. All these variations make it an extremely challenging dataset. We use the standard protocol [12, 16] and report the average performance over the 4 runs, where every time all images of one sample are taken for test while the images of the remaining 3 samples are used for training.

We now analyze the performance of the proposed LHS vs. the LBP/LTP based image representations. Tab. 1 (col. 1 and 2) gives the results for the different methods on the texture datasets. On the texture recognition experiment, i.e. when a sample seen on training is presented for testing with scale and rotation changes, we achieve a near perfect accuracy of 99.5%. Our best method outperforms the best LBP and LTP baselines by 12.2% and 4.5% respectively. We thus conclude that data-adaptive encoding, using higher-order statistics, of local neighborhoods is advantageous when compared to fixed quantization and heuristics as used in LBP and LTP representations. The high accuracy achieved on the texture recognition dataset, Brodatz-32, leads us to conclude that texture recognition, under the presence of rotation and scale variations, can be done almost perfectly.

On the more challenging KTH TIPS 2a dataset for texture categorization, the best performance we obtain is far from saturated at 73%. LHS performs better than LBP and LTP baselines by 3.2% and 1.7% respectively. KTH TIPS 2a dataset has much stronger variations in scale, illumination conditions, pose, etc. than the Brodatz dataset and the experiment is of texture categorization of unseen sample i.e. the test images are of a sample not seen in training. LHS again outperforms LBP/LTP on the task of texture categorization. More recently better results have been reported on the task of texture categorization. It has been demonstrated that standard object image classification pipeline of Fisher Vectors [3, 53] with dense SIFT [30] when applied to texture categorization [29] achieves excellent results. We note that such features are of much higher complexity than the proposed LHS. We analyze LHS wrt. such features in Sec. 5.7, albeit on the task of face verification. Also, it has been shown that representations learnt for image classification tasks, using large amounts of external data, transfer successfully to texture recognition as well [29]. While such methods are quite interesting, they are not directly comparable to the proposed method.

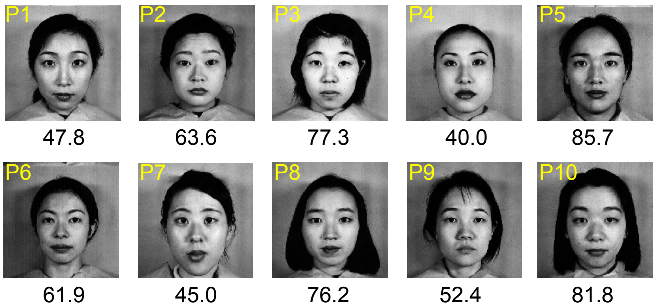

Figure 2: The images of the 10 persons in the neutral expression. The number below is the categorization accuracy for all 7 expressions for the person (see Sec. 5.4).

5.4. Facial analysis

Japanese Female Facial Expressions (JAFFE)[6] [17] is a dataset for facial expression recognition. It contains 10 different females expressing 7 different emotions e.g. sad, happy, angry. We perform expression recognition for both known persons, like earlier works [55], and for unknown person. In the first (experiment E1), one image per expression for each person is used for testing while remaining ones and used for training. Thus, the person being tested is present (different images) in training. In the second (experiment E2), all images of one person are held out for testing while the rest are used for training. Hence, there are no images of the person being tested in the training images, making the task more challenging. For both cases, we report the mean and standard deviation of average accuracies of 10 runs.

Tab. 1 (col. 3 and 4) gives the performance of the different methods on the expression categorization task. On the first experiment (E1) we obtain very high accuracies

[4] http://www.cse.oulu.fi/CMV/TextureClassification

[5] http://www.nada.kth.se/cvap/datasets/kth-tips/

[6] http://www.kasrl.org/jaffe.html

(a) Rectangular sampling (8-pixel neighborhood)

	Brodatz–32	KTH TIPS 2a	JAFFE E1	JAFFE E2
LBP baseline	87.2 ± 1.5	69.8 ± 6.9	86.9 ± 2.6	56.5 ± 21.0
LTP baseline	95.0 ± 0.8	69.3 ± 5.3	93.6 ± 1.8	57.2 ± 16.3
LHS (proposed)	**99.3 ± 0.3**	**71.7 ± 5.7**	**95.6 ± 1.7**	**64.6 ± 19.2**

(b) Circular sampling (bilinear interpolation for diag. neighbproposed)

	Brodatz–32	KTH TIPS 2a	JAFFE E1	JAFFE E2
LBP baseline	87.3 ± 1.5	69.8 ± 6.7	94.3 ± 2.1	61.8 ± 24.1
LTP baseline	94.9 ± 0.8	71.3 ± 6.3	95.1 ± 1.8	60.6 ± 20.8
LHS (proposed)	**99.5 ± 0.2**	**73.0 ± 4.7**	**96.3 ± 1.5**	**63.2 ± 16.5**

Table 1: Results (avg. accuracy and std. dev.) on the different datasets.

as the task is of recognition of expressions, from a never seen image, of a person who was already seen at training. The proposed LHS again outperforms LBP and LTP based representation by 2% and 1.2%, respectively. On the more challenging second experiment (E2), i.e. when the test subject was not seen during training, we see that the accuracies are much less than E1. LHS again outperforms the best LBP and LTP accuracies by 2.8% and 4% respectively. Fig. 2 shows one image of each of the 10 persons in the dataset along with the expression recognition accuracy for that person. This dataset has highly variable intra-person differences i.e. for some individuals different expression images are close while for others they are very different. This results in very different accuracies for the different persons and hence high standard deviation, for all the methods. We conclude that LHS, owing to more accurate description of local pixel neighborhoods, is able to perform better than the LBP/LTP based image description for the task of facial expression categorization on the JAFFE dataset.

Labeled Faces in Wild (LFW) [18] is a popular dataset for face verification by unconstrained pair matching. Face verification is the task where two face images are given and the system has to predict whether they are of the same person or not, with the possibility that the(those) person(s) might not have been seen at training. Hence, it is different from face recognition, where the system has to recognize a person already seen at training. It stresses the system to find characteristics which are general and make the faces similar or not, rather than characteristics which are specific to a known set of persons. LFW contains 13,233 face images of 5749 different individuals of different ethnicity, gender, age, etc. It is an extremely challenging dataset and contains face images with large variations in pose, lighting, clothing, hairstyles, etc. (Fig. 3 shows example pairs from the dataset). LFW dataset is organized into two parts: 'View 1' is used for training and validation (e.g. for choosing the parameters) while 'View 2' is only for final testing and benchmarking. In our setup, we follow the specified training and evaluation protocol. We use the, publicly available, aligned version of the faces as provided by Wolf et al. [56][7].

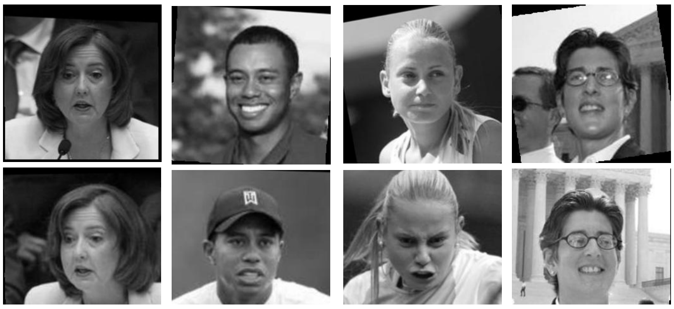

Figure 3: Example image pairs from the LFW [18, 56] dataset. Note the large variation in appearance due to different pose, expression, illumination, accessories etc.

We first report results in the restricted unsupervised task of the LFW dataset, i.e. (i) we use strictly the data provided without any other data from any other source and (ii) we do not utilize class labels while obtaining the image representation. This task evaluates the information contained in the features without help from any supervised modifications. We will provide results later in the supervised setting in Sec. 5.7 where we will demonstrate that, combined with supervised learning, LHS give a very attractive trade-off between performance and speed wrt. the state-of-the-art methods.

We center crop the face images to 150×80 and resize them to 70×40 pixels. We then compute the features with a 7×4 grid, of 10×10 pixels cells, overlayed on the image. We compute the LHS representations for each cell separately and compute the similarity between image pairs as the mean of L2 distances between the representations of corresponding cells. We classify image pairs into same or not same by thresholding on their similarity. We choose the testing threshold as the one which gives the best classification accuracy on the training data.

LHS gives an accuracy of 73.4% with a standard error on the mean of 0.4%. This is a competitive performance in the unsupervised setting for the dataset. Unlike other methods, it neither uses external data e.g. as by PAF [65],

[7] http://www.openu.ac.il/home/hassner/data/lfwa/

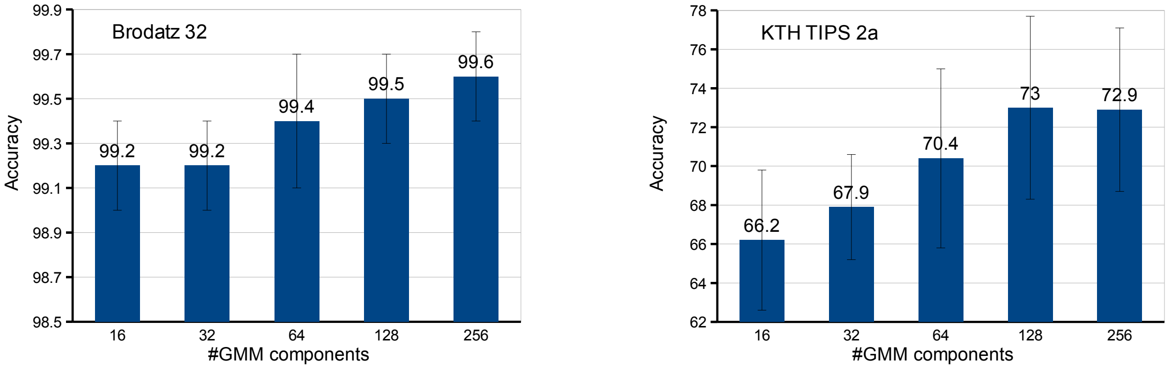

Figure 4: The accuracies of the method for different number of GMM components for Brodatz (left) and KTH TIPS 2a (right) dataset (see Sec. 5.5)

nor does it do feature-specific post-processings e.g. as by LQP [64]. We compare with other existing approaches, including those based on LBP in Sec. 5.6. Also, we show in Sec. 5.7 that LHS is one of the best performing methods, among methods of similar low complexity, in the supervised face verification setting.

5.5. Effect of sampling and number of components

Table 1 shows the results with (a) rectangular 3×3 pixel neighborhood and (b) LBP/LTP like circular sampling of 8 neighbors with the diagonal neighbor values bilinearly interpolated. Performance on the Brodatz-32 dataset is similar for both the samplings while that for KTH and JAFFE datasets differ. In general, the circular sampling performs better for all the datasets. We note that the variations in, and hence difficulty of, the Brodatz-32 dataset is much less compared to the other two datasets and hence images in Brodatz-32 dataset are possibly well represented by either of the two samplings. Thus, we conclude that, in general, circular sampling is to be preferred as it performs better on most of the datasets and generates more discriminative statistics.

Fig. 4 shows the performance on the two texture datasets for different number of mixture model components. The length of the vector, and hence the space and time complexity of the method, varies proportional to the number of components in the GMM. Relatively higher number of components leads to a higher likelihood, i.e. a better fit to the data, and hence a better description of the space but also leads to vectors which are longer to compute and store. On this trade-off of size and accuracy of description, we observe that the performance, for both texture datasets, improves with the number of components and saturates at 128. On the Brodatz-32 dataset, LHS is able to give more than 99% accuracy with just 16 components, highlighting the relatively easier nature of this dataset. While for the KTH dataset, performance improves significantly when the number of components increase from 16 to 128 (by absolute 6.8%). KTH is significantly more challenging than the Brodatz-32 dataset and hence requires more accurate and costly descriptors computed from larger number of mixture components.

5.6. Comparison with existing methods

Table 2 shows the performance of the proposed LHS along with existing methods. On the Brodatz dataset we outperform all methods and to the best of our knowledge report, near perfect, state-of-the-art performance. On the JAFFE dataset, as well, we achieve the best results reported till date. On the KTH dataset, Chen et al. [12] report an accuracy of 64.7% using their Weber law descriptors (WLD) with KNN classifier. Caputo et al. [16] report 71.0% for their 3-scale LBP and *non-linear* chi-squared radial basis function kernel based SVM classifier. In comparison we use linear classifiers which are not only fast to train but also need only a vector dot product at test time (cf. kernel computation with support vectors which is of the order of number of training features). Note Caputo et al. obtain their best results with multi scale features and a complex decision tree (with non-linear classifiers at every node). We expect our features to outperform their features with similar complex classification architecture. The higher complexity Fisher vectors with SIFT features (SIFT-FV) [29] achieves substantially better on the KTH dataset (82.2% vs. 73.0%), however they are orders of magnitude longer and slower than the proposed method. We provide space and time comparisons with the proposed method and SIFT-FV method below Sec. 5.8, Tab. 5, with images from LFW dataset without loss of generality.

Tab. 2(d) reports accuracy rates of our method and those of competing unsupervised methods[8] on LFW dataset. Our method outperforms the LBP baseline (LBP with χ^2 distance) [62] by 3.9% and gives 1.2% better performance than Locally Adaptive Regression Kernel (LARK) features of [63]. The better performance of our features, compared to the LBP baseline and fairly complex LARK features, on this difficult dataset once again underlines the fact that local neighborhood contains a lot

[8] For more results, see webpage http://vis-www.cs.umass.edu/lfw/results.html

(a) Brodatz–32

Method	Acc.	Remark
Jalba et al. [57]	93.5	Morphological hat-transform
Urbach et al. [58]	96.5	Connected shape size pattern spectra
Ojala et al. [59]	96.8	Distributions of signed gray level differences
Chen et al. [12]	97.5	Weber law feat. + k-NN
LHS (proposed)	**99.3**	

(b) JAFFE

Method	Acc.	Remark
Shan et al. [52]	81.0	LBP based
Guo et al. [60]	91.0	Gabor filters + feat. selection
Lyons et al. [61]	92.0	Gabor filters + Linear Discriminant Analysis
Feng et al. [51]	93.8	LBP + Linear programming
LHS (proposed)	**95.6**	

(c) KTH TIPS 2a

Method	Acc.	Remark
Chen et al. [12]	64.7	Weber law feat. + k-NN
Caputo et al. [16]	71.0	3 sc. LBP, nonlin. SVM
LHS (proposed)	73.0	
DeCAF [29]	78.4	Large amount of labeled external data
SIFT-FV [29]	**82.2**	Higher complexity, see § 5.8

(d) LFW (aligned, unsupervised)

Method	Acc.	Remark
Javier et al. [62]	69.5 ±0.5	LBP with χ^2 dist.
Seo et al. [63]	72.2 ±0.5	Locally Adaptive Regression Kernel
LHS (proposed)	73.4 ±0.4	
LQP [64]	75.3 ±0.8	Higher complexity
PAF [65]	**87.8** ±0.5	External data for pose correction

Table 2: Comparison with current methods with comparable experimental setup (reports accuracy, see Sec. 5.6).

of discriminative information. It also demonstrates the representational power of our features, which are successful in encoding the information missed by other methods. More recent works have reported higher performances e.g. Local Quantized Patterns (LQP) [64] achieves 75.3% without any postprocessing and gain even higher when postprocessed with whitened PCA and compared with cosine similarity. However, LQP have higher complexity than LHS and hence gain 2%. While the current LHS features are only 3584 dimensional, the LQP features are 36000 dimensional i.e. 10× longer. Pose Adaptive Filters (PAF) [65] use external data to learn pose robust features using 3D fitting of faces and achieve substantially more. This underlines the fact that the dataset has very challenging pose variations, correcting which will arguably improve the performance of the proposed LHS features as well. Since they use external data while the proposed method does not, their performance is not directly comparable to that of the proposed method. Also, adding pose robustness with additional effort, e.g. 3D fitting of face and using external data, is another challenging problem in itself and has not been explored further in this work.

Thus, we conclude, the proposed method is capable of achieving competitive results while being computationally simple and efficient.

5.7. *LHS with supervised discriminative metric learning*

We now provide results of the proposed Local Higher-order Statistics (LHS) features with supervised discriminative metric learning (ML) on the challenging Labeled Faces in the Wild (LFW) [18] dataset. We show that when used with such supervised ML, which can be equivalently seen as a projection to a lower dimensional discriminative subspace (see Sec. 4), the LHS features can obtain very high performance while being much more efficient than the competition.

We operate in the *'Supervised, unrestricted, label-free outside data'* protocol. Tab. 3 gives the performance of LHS for different values of the parameters. We see that the increasing the number of Gaussian components steadily increases the performance from $k = 4$ to $k = 24$ by a little less than 2% absolute while beyond that the results seem to saturate. Similarly, for a fixed number of Gaussian components, increasing the projection dimension increases the results but with a pronounced diminishing returns effect.

It is quite interesting to note these performances in the context of existing methods. Tab. 4 shows the performance of LHS wrt. state-of-the-art methods on LFW dataset. LFW achieves the best results among the features in the low complexity regime, and competitive results among features with high complexity or methods that combine multiple features. In particular our own implementation of Local Binary Patterns (LBP) using the (default parameters of the) `vlfeat` library [54] gives 86.2% with a feature

Supervised, unrestricted, label free outside data

#Gauss.	Dimension		ROC-EER
(k)	org. (d_0)	proj. (d)	Accuracy
4	1792	32	85.73 ± 0.17
		64	86.37 ± 0.19
		128	**86.60** ± **0.17**
8	3584	32	86.47 ± 0.17
		64	87.37 ± 0.14
		128	**87.60** ± **0.14**
16	7168	32	87.43 ± 0.17
		64	87.57 ± 0.17
		128	**88.13** ± **0.15**
24	10752	32	87.63 ± 0.17
		64	87.93 ± 0.20
		128	**88.27** ± **0.17**
32	14336	32	87.47 ± 0.20
		64	**88.03** ± **0.16**
		128	**87.97** ± **0.14**

Table 3: Results of proposed LHS on the Labeled Faces in the Wild (LFW) [18] dataset for different parameter settings.

Method	Space		Time	
	dim.	reduction	ms	speedup
SIFT-FV	67584	Ref.	2400*	Ref.
LHS	1792	38×	13	185×
	3584	19×	15	160×
	7168	9×	19	126×
	10752	6×	22	109×
	14336	5×	25	96×

Table 5: The space and time complexity comparison between proposed LHS the FV method. (*) The time for the best performing configuration in [49], i.e. step size 1, is interpolated from the time reported for step size 2 (0.6s). Our implementation of fisher vectors takes similar time, see Sec. 5.8

dimension of 7k. Compared to this LHS with only 1k dimensions gives 86.6% (Tab. 3) and that with 10k dimension gives 88.3%. When combined with LBP the performance increases to 89.0%. Our implementation of Fisher vectors with dense SIFT features gives 92.9% (compared to 93.0% reported in [49]), and when combined with LHS the performance improves to 93.5%, which is a modest improvement over the state-of-the-art in the *'Supervised, unrestricted, label-free outside data'* protocol[9]. Thus, we conclude that LHS features are competitive in the low complexity domains and are complementary to the high complexity features for supervised face verification on LFW. In the next section we discuss their time and space benefit over the high complexity features.

Methods with similar complexity

Method	Accuracy
LBP + ITML [66]	85.1 ± 0.6
LBP + PLDA [67]	87.3 ± 0.6
LHS + JML (proposed)	**88.2** ± **0.2**

Methods with multiple feats/higher complexity

Method	Accuracy
comb. LDML-MkNN [47]	87.5 ± 0.4
comb. PLDA [67]	90.1 ± 0.5
SIFT-FV [49]	93.0 ± 1.1
High dim LBP [68]	93.2 ± 1.1
LBP + LHS (proposed)	89.0 ± 0.1
SIFT-FV + LHS (proposed)	**93.5** ± **0.2**

Methods using large amts of external labeled data

Method	Accuracy
High dim LBP [68]	95.2 ± 1.1
Deep learning [36, 38]	**97.4** ± **0.3**

Table 4: Comparison with existing works on the Labeled Faces in the Wild (LFW) [18] dataset–unrestricted and supervised setting.

5.8. Time and space complexity of LHS

The proposed LHS features are very compact and efficient to compute. Compared to one of the state-of-the-art systems for face verification [49] they are about two orders of magnitude faster and an order of magnitude smaller. Tab. 5 gives the space and computation time comparison of the LHS features wrt. Fisher vectors with SIFT features (SIFT-FV) [49]. The experiments were run on a server with 2.67 GHz Intel Xeon processor running Ubuntu 14.04 and all the data was loaded in RAM for timing the computations. The times reported are for a single threaded program using one core.

The best performing LHS features are 10,752 dimensional and take 22 ms to compute compared to 67,584 for SIFT-FV which amounts to a space saving of 6× and speedup of 109×; while the most lightweight LHS configuration tested is 38× smaller and 185× faster than SIFT-FV. Ignoring the offline training time, which is $O(d_0^2)$, and considering only the online testing times, the best performance is reached when the image pairs are horizontally flipped and the distance between the four combinations are averaged. Hence, for comparing a face pair, features for 4 images need to be calculated i.e. Fisher vectors take 9.6s while the proposed LHS take only 88ms, both on a single core of a modern CPU. Such advantages come with a drop in performance, but might be essential for time and space critical applications e.g. in embedded systems. They might also be used in a cascade system where the efficient LHS features are used to tackle the easy decisions while delegating the tougher examples to the higher complexity features, thereby reducing the average time over several comparisons.

We note that, our implementation of LHS is in unopti-

[9] For more results, see webpage http://vis-www.cs.umass.edu/lfw/results.html

mized C/C++, called via the MEX interface of MATLAB. Arguably it can be improved substantially, in particular, by tuning/approximating the GMM posterior probability estimation, which involves costly exponential operations.

6. Conclusions

We have presented a model that captures higher-order statistics of small local neighbohoods to produce a highly discriminative representation of the images. Our experiments, on two challenging texture datasets and two challenging facial analysis datasets, validate our approach and show that the proposed model encodes more local information than the competing methods and achieves competitive results. Two of the datasets we used, one each for textures and faces, were relatively simpler while two other were more difficult. The results on the simpler datasets served to demonstrate that the method is capable of having a richer appearance descriptor compared to existing methods. While on the more challenging case, we showed with experiments on the supervised task of face verification on the challenging Labeled Faces in the Wild (LFW) dataset that the proposed method achieves best results for low complexity features and is complementary to the high dimensional features. When combine with the state-of-the-art method it improves the performance to establish a new state-of-the-art on the LFW dataset when no external labeled data is used. Compared to the best method the proposed method is two orders of magnitude faster to compute and an order of magnitude compact making it a very appropriate choice for low complexity devices e.g. embedded systems.

While the current state-of-the-art systems, based on deep networks trained with large amounts of external data [32, 33, 34, 35, 36, 37, 38, 39, 40, 41] have gained, the proposed method is still relevant due to its speed – we could use it as an initial low complexity stage of a cascade based system. Also, in very low complexity/cost systems the size of the model might also limit the use of deep networks and make the proposed method relevant.

Acknowledgments. The authors acknowledge support from the PHYSIONOMIE project, grant number ANR-12-SECU-0005-01.

References

[1] M. Pietikinen, A. Hadid, G. Zhao, T. Ahonen, Computer Vision Using Local Binary Patterns, Springer, 2011. 1, 2, 3, 4

[2] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: Application to face recognition, PAMI 28 (12). 1, 2, 3

[3] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: NIPS, 1999. 2, 3, 4, 6, 7

[4] J. Puzicha, T. Hofmann, J. M. Buhmann, Non-parametric similarity measures for unsupervised texture segmentation and image retrieval, in: CVPR, 1997. 1, 2

[5] Y. Rubner, C. Tomasi, Texture-based image retrieval without segmentation, in: ICCV, 1999. 1, 2

[6] T. J. Leung, J. Malik, Representing and recognizing the visual appearance of materials using three-dimensional textons, IJCV 43 (2001) 29–44. 1, 2, 3

[7] O. G. Cula, K. J. Dana, Compact representation of bidirectional texture functions, in: CVPR, 2001. 1, 2

[8] S. C. Zhu, Y. Wu, D. Mumford, Filters, random-fields and maximum-entropy (FRAME): Towards a unified theory for texture modeling, IJCV 27 (1998) 107–126. 1, 2

[9] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, PAMI 24 (7) (2002) 971–987. 1, 2, 3, 5, 6

[10] M. Varma, A. Zisserman, Texture classification: Are filter banks necessary?, in: CVPR, 2003. 1, 2, 3

[11] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, TIP 19 (6) (2010) 1635–1650. 1, 3, 5, 6

[12] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, W. Gao, WLD: A robust local image descriptor, PAMI 32 (9) (2010) 1705–1720. 1, 3, 7, 9, 10

[13] G. Sharma, S. U. Hussain, F. Jurie, Local higher-order statistics (LHS) for texture categorization and facial analysis, in: ECCV, 2012. 2

[14] K. Valkealahti, E. Oja, Reduced multidimensional co-occurence histograms in texture classification, PAMI 20 (1) (1998) 90–94. 2, 7

[15] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover Publications, New York, 1966. 2, 7

[16] B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, in: ICCV, 2005. 2, 7, 9, 10

[17] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: AFGR, 1998. 2, 7

[18] G. B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: A database for studying face recognition in unconstrained environments, Tech. Rep. 07-49, University of Massachusetts, Amherst (October 2007). 2, 8, 10, 11

[19] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: ICCV, 2003. 3

[20] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Intl. Workshop on Stat. Learning in Comp. Vision, 2004. 3

[21] L. Liu, P. Fieguth, G. Kuang, Compressed sensing for robust texture classification, in: ACCV, 2010. 3

[22] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, PAMI 27 (2005) 1265–1278. 3

[23] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, IJCV 73 (2) (2007) 213–238. 3

[24] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, IJCV 62 (2005) 61–81. 3

[25] M. Croiser, L. D. Griffin, Using basic image features for texture classification, IJCV 88 (2010) 447–460. 3

[26] E. Hayman, B. Caputo, M. Fritz, J.-O. Eklundh, On the significance of real world conditions for material classification, in: ECCV, 2004. 3

[27] Y. Xu, H. Ji, C. Fermuller, View point invariant texture description using fractal analysis, IJCV 83 (2009) 85–100. 3

[28] Y. Xu, X. Yang, H. Ling, H. Ji, A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid, in: CVPR, 2010. 3

[29] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: CVPR, 2014. 3, 7, 9, 10

[30] D. Lowe, Distinctive image features form scale-invariant keypoints, IJCV 60 (2) (2004) 91–110. 3, 7

[31] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, in: ICML, 2014. 3

[32] C. Nhan Duong, K. Luu, K. Gia Quach, T. D. Bui, Beyond principal components: Deep boltzmann machines for face modeling, in: CVPR, 2015. 3, 12

[33] G. B. Huang, H. Lee, E. Learned-Miller, Learning hierarchi-

cal representations for face verification with convolutional deep belief networks, in: CVPR, 2012. 3, 12

[34] J. Hu, J. Lu, Y.-P. Tan, Discriminative deep metric learning for face verification in the wild, in: CVPR, 2014. 3, 12

[35] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: CVPR, 2015. 3, 12

[36] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: CVPR, 2014. 3, 11, 12

[37] Y. Sun, X. Wang, X. Tang, Deeply learned face representations are sparse, selective, and robust, in: CVPR, 2015. 3, 12

[38] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: CVPR, IEEE, 2014. 3, 11, 12

[39] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Web-scale training for face identification, in: CVPR, 2015. 3, 12

[40] M. Kan, S. Shan, H. Chang, X. Chen, Stacked progressive autoencoders (SPAE) for face recognition across poses, in: CVPR, 2014. 3, 12

[41] Z. Zhu, P. Luo, X. Wang, X. Tang, Deep learning identity-preserving face space, in: ICCV, 2013. 3, 12

[42] A. M. Martínez, Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class, PAMI 24 (6) (2002) 748–763. 3

[43] C. M. Bishop, Pattern recognition and machine learning, Springer, 2006. 4

[44] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, in: ECCV, 2010. 4

[45] A. Vedaldi, A. Zisserman, Efficient additive kernels using explicit feature maps, in: CVPR, 2010. 4, 7

[46] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, in: arXiv:1306.6709, 2013. 5

[47] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: ICCV, 2009. 5, 11

[48] A. Mignon, F. Jurie, PCCA: A new approach for distance learning from sparse pairwise constraints, in: CVPR, 2012. 5

[49] K. Simonyan, O. M. Parkhi, A. Vedaldi, A. Zisserman, Fisher vector faces in the wild, in: BMVC, 2013. 5, 6, 11

[50] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: ECCV, 2012. 5

[51] X. Feng, M. Pietikinen, T. Hadid, Facial expression recognition with local binary patterns and linear programming, Pattern Recognition and Image Analysis 15 (2005) 546–548. 6, 10

[52] C. Shan, S. Gong, P. W. McOwan, Facial expression recognition based on local binary patterns: A comprehensive study, IVC 27 (2009) 803–816. 6, 10

[53] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice, IJCV 105 (3) (2013) 222–245. 6, 7

[54] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, http://www.vlfeat.org/ (2008). 6, 10

[55] S. Liao, W. Fan, A. C. Chung, D. Yan Yeung, Facial expression recognition using advanced local binary patterns, tsallis entropies and global appearance features, in: ICIP, 2006. 7

[56] L. Wolf, T. Hassner, Y. Taigman, Similarity scores based on background samples, in: ACCV, 2009. 8

[57] A. C Jalba, M. HF Wilkinson, J. BTM Roerdink, Morphological hat-transform scale spaces and their use in pattern classification, PR 37 (5) (2004) 901–915. 10

[58] E. R. Urbach, J. B. Roerdink, M. H. Wilkinson, Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images, PAMI 29 (2) (2007) 272–285. 10

[59] T. Ojala, K. Valkealahti, E. Oja, M. Pietikäinen, Texture discrimination with multidimensional distributions of signed gray-level differences, PR 34 (3) (2001) 727–739. 10

[60] G. Guo, C. R. Dyer, Simultaneous feature selection and classifier training via linear programming: A case study for face expression recognition, in: CVPR, 2003. 10

[61] M. J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, PAMI 21 (12) (1999) 1357–1362. 10

[62] J. Ruiz-del Solar, R. Verschae, M. Correa, Recognition of faces in unconstrained environments: a comparative study, EURASIP Journal on Advances in Signal Processing 2009. 9, 10

[63] H. J. Seo, P. Milanfar, Face verification using the LARK representation, Information Forensics and Security, IEEE Transactions on 6 (4) (2011) 1275–1286. 9, 10

[64] S. U. Hussain, T. Napoléon, F. Jurie, et al., Face recognition using local quantized patterns, in: BMVC, 2012. 9, 10

[65] D. Yi, Z. Lei, S. Z. Li, Towards pose robust face recognition, in: CVPR, IEEE, 2013. 8, 10

[66] Y. Taigman, L. Wolf, T. Hassner, Multiple one-shots for utilizing class label information, in: BMVC, 2009. 11

[67] P. Li, Y. Fu, U. Mohammed, J. H. Elder, S. J. Prince, Probabilistic models for inference about identity, PAMI 34 (1) (2012) 144–157. 11

[68] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification, in: CVPR, 2013. 11

Gaurav Sharma is currently at the Max Planck Institute for Informatics, Germany. He holds an Integrated Master of Technology (5 years programme) in Mathematics and Computing from the Indian Institute of Technology Delhi (IIT Delhi) and a PhD in Applied Computer Science from INRIA (LEAR team) and the Université de Caen Basse-Normandie, France. His primary research interest lies in Machine Learning applied to Computer Vision tasks such as image classification, object recognition and facial analysis.

Frédéric Jurie is a professor at the French Université de Caen Basse-Normandie (GREYC - CNRS UMR6072) and an associate member of the INRIA-LEAR team. His research interests lie predominately in the area of Computer Vision, particularly with respect to object recognition, image classification and object detection.