Local Higher-Order Statistics (LHS) – A Novel Image Representaion for Texture Categorization and Facial Analysis

Gaurav Sharma, Frédéric Jurie

Local Higher-Order Statistics (LHS) – A Novel Image Representaion for Texture Categorization and Facial Analysis

Gaurav Sharma, Frédéric Jurie

GREYC CNRS UMR 6072, Université de Caen Basse-Normandie, France

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. In contrast with models based on the global structure of textures and faces, it has been recently shown that small local pixel pattern distributions can be highly discriminative while being extremely efficient to compute. Motivated by such works, the proposed model employs higher-order statistics of local non-binarized pixel patterns for the image description. Hence, in addition to being remarkably simple, it requires neither any user specified quantization of the space (of pixel patterns) nor any heuristics for discarding low occupancy volumes of the space. 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 outperforming contemporary methods, with similar complexity setup. Further, it is complementary to higher complexity features and when combined with them improves performance.

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

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. Texture recognition is beneficial for many 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, as the models developed for texture recognition are generally found to be competitive for face analysis. Analysis of faces, similarly, finds important applications 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.

Earlier works on texture analysis were focused on the development and application of filter banks *e.g.* [1, 2, 3]. They computed filter response coefficients for a number of filters or wavelets and learned their distributions. However, later works disproved the necessity of such ensembles of filters *e.g.* Ojala *et al.* [4] and Varma and Zisserman [5] showed that it is possible to discriminate between textures using pixel neighbourhoods as small as 3×3 pixels. They demonstrated that despite the global structure of the textures, very good discrimination could be achieved by exploiting the distributions of such small pixel neighbourhoods. More recently, exploiting such *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 [6, 7, 8, 9] for systems with low complexity. However, as we discuss later, these methods suffer from several important limitations, such as the use of fixed quantization of the feature space as well as the use of heuristics to prune volumes in the feature space. In addition, they represent feature distributions with histograms and hence are restricted to the use of low order statistics.

In contrast to these previous works, we propose a model that represents images with higher order statistics of local pixel neighbourhoods. We obtain a data driven partition of the feature space using parametric mixture models, to represent the distribution of the vectors, and learn the parameters from the training data. Hence, the coding of vectors is intrinsically adapted to any classification task and the computations involved remain very simple despite the strengths. We validate our approach by extensive experiments on four challenging datasets: (i) Brodatz 32 texture dataset [10, 11], (ii) KTH TIPS 2a materials dataset [12], (iii) Japanese Female Facial Expressions (JAFFE) dataset [13], and (iv) Labeled Faces in the Wild (LFW) dataset [14], and show that using higher-order statistics gives a more expressive description and leads to state-of-the-art performance in low complexity settings. We also show that they are complementary to the recent high complexity state-of-the-art methods and, in partic-

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

ular, we show that their combination with such methods gives the state-of-the-art performance on the very challenging LFW dataset in the unsupervised protocol, when no external labeled data is used.

This paper extends the work of Sharma *et al.* [15] with a better description of the method, improved discussion *wrt.* the current state-of-the-art methods and thorough experimental results, particularly in the case of supervised face verification task. The proposed Local Higher-order Statistics (LHS) are shown to be highly effective and efficient, when combined with state-of-the-art supervised metric learning methods.

1.1. Related works

Most of the earlier works on texture analysis focused on the development of filter banks and on characterizing the statistical distributions of their responses e.g. [1, 2, 3], until Ojala *et al.* [4] and, more recently, Varma and Zisserman [5] showed that statistics of small pixel neighbourhoods are capable of achieving high discrimination. Since then many methods working with local pixel neighbourhoods have been used successfully in texture and face analysis, *e.g.* [8, 9, 16].

Local pixel pattern operators, such as Local Binary Patterns (LBP) by Ojala *et al.* [4], have been very successful for local pixel neighbourhood description. LBP based image representation aims to capture the joint distribution of local pixel intensities. LBP approximates the distribution by first taking the differences between the center pixel and its neighbours and then considering just the signs of the differences. The first approximation lends invariance to gray-scale shifts and the second to intensity scaling. Local Ternary Patterns (LTP) were introduced by Tan and Triggs [8] to add resistance to noise. LTP requires a 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. 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. The use of these patterns is motivated by the empirical observation that uniform patterns account for nearly 90 percent of all observed patterns in textures. Although it works quite well in practice, still it is a heuristic for discarding low occupancy volumes in feature space.

Most of the other recent methods, driven by the success of earlier texton based texture classification method [1] and recent advances in the field of object category classification, adopt bag-of-words models to represent textures as distributions of local textons [5, 16, 17, 18, 19, 20, 21, 22, 23]. They learn a dictionary of textons obtained by clustering vectors (*e.g.* based on either pixel intensities, sampled on local neighbourhoods, or their differences), and then represent the image as histograms over the learnt codebook vector assignments. The local vectors are derived in multiple ways, incorporating different invariances like rotation, view point *etc. E.g.* [17, 18] generate an image specific texton representation from rotation and scale invariant descriptors and compare them using Earth Movers distance, whereas [5, 4, 16, 19] use a dictionary learned over the complete dataset to represent each image as histogram over this dictionary.

In a more recent work, traditional image classification methods when applied to the more challenging textures in the wild scenario have shown very good performances [24]. Such methods use classic local features such as SIFT [25] with different encoding methods, particularly Fisher scores [26], similar to those employed in the present work. They [24] also evaluate deep learning [27] based representation and show 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.

Motivations. The motivations for this paper follow the conclusions that can be drawn from these related works. (i) As shown by [4, 5], and by all the recent papers that build on these, modeling distributions of small pixel neighbourhoods (as small as 3×3 pixels) can be very effective. (ii) Unfortunately, all the previously mentioned related approaches involve coarse approximations that prevent them from getting all the benefits of an accurate representation of such small neighbourhoods, and (iii) all these methods use low-order statistics, generally zeroth order counts *i.e.* histograms, while using high-order moments can give a more expressive representation. Addressing these limitations by accurately describing small neighbourhoods with their higher-order statistics, without coarse approximations, is the main contribution of the present paper.

2. The Local Higher-order Statistics (LHS) Model

As explained before, the proposed Local Higher-order Statistics (LHS) model intends to represent images by exploiting, as well as possible, the distribution of local pixel neighbourhoods. Thus, we start with small pixel neighbourhoods of 3×3 pixels and model the statistics of their local differential vectors.

2.1. Local differential vectors.

We work with all possible 3×3 neighbourhoods in the 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-neighbours. We are interested in exploiting the distribution $p(\mathbf{v}^n|I)$ of the these vectors, for a given image, to represent the image. We obtain invariance to monotonic changes in gray levels by subtracting the value

of the center pixel from the rest and using the difference vector *i.e.*

$$p(\mathbf{v}^n|I) \approx p(\mathbf{v}|I) \tag{2}$$

$$\mathbf{v} = (v_1 - v_c, \ldots, v_8 - v_c). \tag{3}$$

We call the vectors $\{\mathbf{v}\}$ thus obtained as the differential vectors.

2.2. Higher order statistics.

The key contribution of LHS is to use the statistics of the differential vectors $\{\mathbf{v}|\mathbf{v} \in I\}$ to characterize the images. Instead of using a hard and/or predefined quantization, we use parametric Gaussian mixture model (GMM) to derive a probabilistic representation of the differential space. Defining such soft quantization, which can equivalently be seen as a generative model on the differential vectors, allows us to use a characterization method which exploits higher order statistics. We use the *Fisher score* method (Jaakkola and Haussler [26]), where given a parametric generative model, a vector can be characterized by the gradient with respect to the parameters of the model. The Fisher score, for an observed vector $\mathbf{v}$ *wrt.* a distribution $p(\mathbf{v}|\boldsymbol{\lambda})$, where $\boldsymbol{\lambda}$ is parameter vector, is given as,

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

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

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

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

the Fisher scores can be computed using the following partial derivatives (we assume diagonal Σ to decrease the number of parameters to be learnt)

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

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

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

where the square of a vector is element-wise one. In the derivatives above we can see that the information based on the first and second powers of the differential vectors are also coded; these are higher order statistics for the differential vectors. After obtaining the differential vectors corresponding to every pixel neighbourhood in the image, we compute the image representation as the average vector over all of them. 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 [28]. We also incorporate two normalizations (on image vector $\mathbf{x}$) [29] i.e. power normalization,

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

and L2 normalization,

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

Perronnin *et al.* [29] motivate the power normalization for obtaining a *de-sparsification* effect. Similar power normalization has also been shown as an *explicit feature map* by Vedaldi and Zisserman [30] *i.e.* a mapping which transforms the vectors to a space where the dot product of the transformed vectors corresponds to the Bhattacharyya 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 metric (detailed below in Sec. 3) for supervised pair matching *i.e.* verification task.

Algorithm 1 Computing Local Higher-order Statistics

1: Randomly sample 2×2 pixels differential vectors $\{\mathbf{v} \in I|I \in \mathcal{I}_{train}\}$
2: Learn the GMM parameters $\{\alpha_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k|k = 1 \ldots K\}$ with EM algorithm on $\{\mathbf{v}\}$
3: Compute the higher-order Fisher scores for $\{\mathbf{v}\}$ using Eq. (7)
4: Compute means C_μ^i and variances C_Σ^i for each coordinate $i \in \{1, \ldots, d_0\}$
5: **for all** images $\{I\}$ **do**
6: Compute all differential vectors $\mathbf{v} \in I$
7: Compute the Fisher scores for all features $\{\mathbf{v}\}$ using Eq. (7)
8: Compute the image representation $\mathbf{x}$ as the average score over all features
9: Normalize each coordinate i as $x^i \leftarrow (x^i - C_\mu^i)/C_\Sigma^i$
10: Apply normalizations, Eq. (8) and (9)
11: **end for**

2.3. Relation to LBP/LTP.

We can view LHS vectors as generalization of local binary/ternary patterns (LBP/LTP) [4, 8]. In LBP every pixel is coded as a binary vector of 8 bits with each bit indicating whether each of the neighbouring 8 pixels, in the 3×3 patch centered on the current pixel, is of greater intensity than the current pixel or not. We can derive the LBP [4] 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. 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 [8] and so on. The use of *uniform patterns* (patterns with exactly one 0-1 and one 1-0 transitions), in both LBP/LTP, can be seen as an empirically derived heuristic for ignoring volumes in differential space which have low occupancies. Thus, the binary/ternary patterns are obtained with a quantization step and rejection heuristic while in our case similar information is learnt from data.

3. 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 discrimnation 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 [31]. More closely related to the 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 [32, 33, 34, 35]. Inspired by such works we now present the method we use to learn a metric using the proposed LHS face representation.

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

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

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

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

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 $d \ll d_0$[1] dimensional vectors.

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 $i = 1, \ldots,$`niters` **do**
  Randomly sample a face pair $(\mathbf{x}_i, \mathbf{x}_j, y_{ij})$ from $\mathcal{T}$
  Compute $D_J^2(\mathbf{x}_i, \mathbf{x}_j)$ using Eq. 10
  if $y_{ij}(b - D_J^2(\mathbf{x}_i, \mathbf{x}_j)) < m$ **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

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

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

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.

4. Experimental Results

The experimental validation is done on four challenging publicly available datasets of textures and faces. We first discuss implementation details then present the datasets and finally give the experimental results for each dataset.

As our focus is on the rich and expressive representation of local neighbourhoods, 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 features.

4.1. Implementation details.

We use only the intensity information of the images and convert color images, if any, to grayscale. We consider two neighbourhood sampling strategies (i) rectangular sampling, where the 8 neighbouring pixels are used, and (ii) circular sampling, where, like in LBP/LTP [4, 8], we interpolate the diagonal samples to lie on a circle, of radius one,

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

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. 4.5). 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 [36, 37]. We crop the 256×256 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. 3. 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 4 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) [34, 38, 26]. 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 [39], 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 $2 \times k \times d_s \times N_c = 57344$.

4.2. Baselines.

We consider baselines of single scale LBP/LTP features generated using the same samplings as our LHS features. We use histogram representation over uniform LBP/LTP features. We L1 normalize the histograms and take their square roots and use them with linear SVM. 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 [30]. Thus using linear SVM with square root of histograms is equivalent to SVM with non linear Bhattacharyya kernel. Hence, our baselines are strong baselines.

4.3. Texture categorization

Brodatz – 32 Textures dataset[2] [10, 11] 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 [10] 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 [9], of randomly splitting the dataset into two halves for training and testing, and report average performance over 10 random splits.

KTH TIPS 2a dataset[3] [12] is a dataset for material categorization. It contains 11 materials *e.g.* cork, wool, linen, with images of 4 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 [9, 12] 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.

Tab. 1 (col. 1 and 2) shows the results for the different methods on these texture datasets. We achieve a near perfect accuracy of 99.5% on the Brodatz dataset. Our best method outperforms the best LBP and LTP baselines by 12.2% and 4.5% respectively and demonstrates the advantage of using rich, higher-order, data-adaptive encoding of local neighbourhoods compared to fixed quantization based LBP and LTP representations. Brodatz dataset contains texture images with scale and rotation variations, hence, the high accuracy achieved on the dataset leads us to conclude that texture recognition can be done almost perfectly under the presence of rotation and scaling variations.

On the more challenging KTH TIPS 2a dataset, the best performance we obtain is far from saturated at 73%. The gain in accuracy over LBP and LTP is 3.2% and 1.7% respectively. This 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. Our descriptor again outperforms LBP/LTP and demonstrates its higher discrimination power and the generalization capability. More recently it has been demonstrated that standard object image classification pipeline of Fisher Vectors [38, 26] with dense SIFT [25] when applied to texture categorization [24] achieves excellent results. We note that such features are of much higher complexity than the proposed LHS. We analyse LHS *wrt.* such features in Sec. 4.7, albeit on

[2] http://www.cse.oulu.fi/CMV/TextureClassification
[3] http://www.nada.kth.se/cvap/datasets/kth-tips/

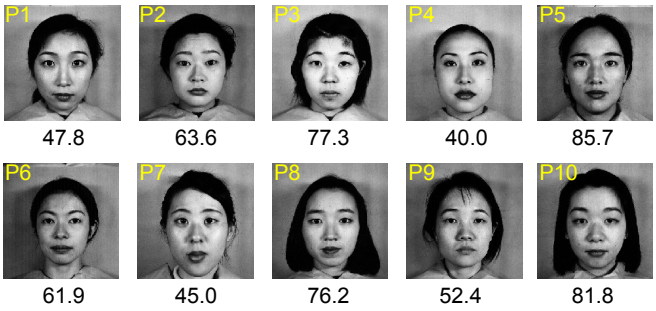

Figure 1: The images of the 10 persons in the neutral expression. The number below is the categorization accuracy for all 7 expressions for the person (see Sec. 4.4).

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 [24].

4.4. Facial analysis

Japanese Female Facial Expressions (JAFFE)[4] [13] 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 [40], 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 image) 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) shows the performance of the different methods. On the first experiment (E1) we obtain very high accuracies as the task is of recognition of expressions, from a never seen image, of a person present in the training set. Our method again outperforms LBP and LTP based representation by 2% and 1.2% respectively. On the more challenging second experiment (E2) we see that the accuracies are much less than E1. Our best accuracy is again better than the best LBP and LTP accuracies by 2.8% and 4% respectively. Fig. 1 shows one image of each of the 10 persons in the dataset along with the expression recognition accuracy for that person. We can see the very high intra-person differences in this dataset, which results in very different accuracies for the different persons and hence high standard deviation, for all the methods.

Labeled Faces in Wild (LFW) [14] is a popular dataset

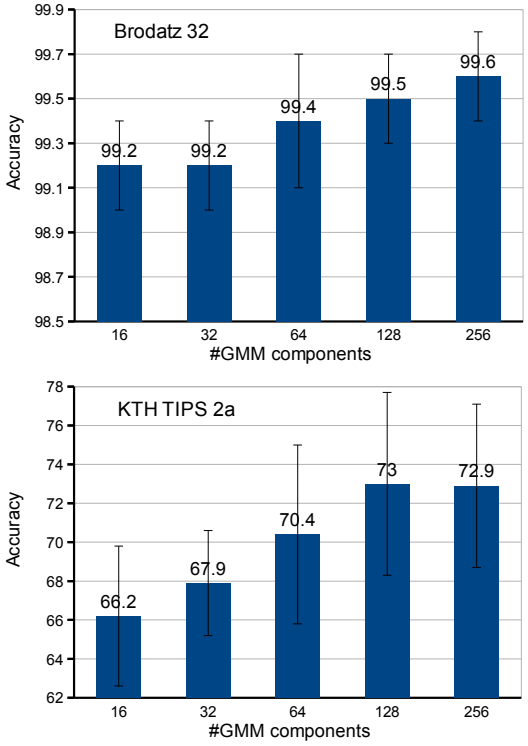

Figure 2: The accuracies of the method for different number of GMM components for Brodatz (left) and KTH TIPS 2a (right) dataset (see Sec. 4.5)

for face verification by unconstrained pair matching *i.e.* given two real-world face images decide whether they are of the same person or not. 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.* LFW dataset is organized into two parts: 'View 1' is used for training, 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.* [41][5].

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. 4.7 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 250×250 images, provided in the

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

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

(a) Rectangular sampling (8-pixel neighbourhood)

	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.

dataset, 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. We obtain 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, while neither using external data *e.g.* by PAF [50], nor doing specific feature post-processings *e.g.* by LQP [49]. We compare with other approaches, including those based on LBP in Sec. 4.6. Also, we show in Sec. 4.7 that LHS is among the best performing methods, of comparable complexity, in the supervised face verification setting.

4.5. Effect of sampling and number of components

Tab. 1 gives the results with (a) rectangular 3×3 pixel neighbourhood and (b) LBP/LTP like circular sampling of 8 neighbproposed, where the diagonal neighbour values are obtained by bilinear interpolation. Performance on the Brodatz dataset is similar for both the samplings while that for KTH and JAFFE datasets differ. In general, the circular sampling seems to be better for all the methods. We note that the variations and difficulty of Brodatz dataset are much less than the other two datasets and hence is possibly well represented by either of the two samplings. Thus, we conclude that, in general, circular sampling is to be preferred as it seems to generate more discriminative statistics.

Fig. 2 shows the performance on the two texture datasets for different number of mixture model components. As this number increases the vector length increases proportionally. Although lower number of components lead to a compact representation, larger numbers lead to better quantization of the space and hence more discriminative representations. We observe that the performance, for both the datasets, increases with the number of components and seems to saturate after a value of 128. Hence, we report results for 128 components. For Brodatz dataset, we see that even with only 16 components the method is able to achieve more than 99% accuracy, highlighting the fewer variations in the dataset. For the KTH dataset we gain significantly by going from 16 to 128 components (6.8 points) which suggests that for more challenging tasks a more descriptive representation is beneficial.

4.6. Comparison with existing methods

Tab. 2 shows the performance of our method 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. Similarly, on the JAFFE and LFW datasets we achieve the best results reported till date.

On the KTH dataset, Chen *et al.* [9], for their recently proposed Weber law based features, report an accuracy of 64.7% with KNN classifier. Caputo *et al.* [12] report 71.0% for their 3-scale LBP and *non-linear* chi-squared RBF 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.

Tab. 2 (d) reports accuracy rates of our method and those of competing unsupervised methods[6] on LFW dataset. Our method not only outperforms the LBP baseline (LBP with χ^2 distance) [47] by 3.9% but also gives 1.2% better performance than current state-of-the-art Locally Adaptive Regression Kernel (LARK) features of [48]. 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 neighbourhood contains a lot of discriminative information. It also demonstrates the representational power of our features which are successful in encoding the information which is missed by other methods. More recent

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

(a) Brodatz–32

Method	Acc.	Remark
Jalba *et al.* [42]	93.5	Morphological hat-transform
Urbach *et al.* [43]	96.5	Connected shape size pattern spectra
Ojala *et al.* [44]	96.8	Distributions of signed gray level differences
Chen *et al.* [9]	97.5	Weber law feat. + k-NN
LHS (proposed)	**99.3**	

(b) JAFFE

Method	Acc.	Remark
Shan *et al.* [37]	81.0	LBP based
Guo *et al.* [45]	91.0	Gabor filters + feat. selection
Lyons *et al.* [46]	92.0	Gabor filters + Linear Discriminant Analysis
Feng *et al.* [36]	93.8	LBP + Linear programming
LHS (proposed)	**95.6**	

(c) KTH TIPS 2a

Method	Acc.	Remark
Chen *et al.* [9]	64.7	Weber law feat. + k-NN
Caputo *et al.* [12]	71.0	3 sc. LBP, nonlin. SVM
LHS (proposed)	73.0	
DeCAF [24]	78.4	Large amount of labeled external data
SIFT-FV [24]	**82.2**	Higher complexity, see § 4.8

(d) LFW (aligned, unsupervised)

Method	Acc.	Remark
Javier *et al.* [47]	69.5 ±0.5	LBP with χ^2 dist.
Seo *et al.* [48]	72.2 ±0.5	Locally Adaptive Regression Kernel
LHS (proposed)	73.4 ±0.4	
LQP [49]	75.3 ±0.8	Higher complexity
PAF [50]	**87.8 ±0.5**	External data for pose correction

Table 2: Comparison with current methods with comparable experimental setup (reports accuracy, see Sec. 4.6).

works have reported higher performances *e.g.* Local Quantized Patterns (LQP) [49] achieves 75.3 without any post-processing and gain even higher when postprocessed with whitened PCA and compared with cosine similarity. Pose Adaptive Filters (PAF) [50] 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.

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

4.7. 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) [14] 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. 3), 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 gaussians, 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 [39] gives 86.2% with a feature 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 [34]), and when combined with LHS the performance improves to 93.5%, which is a modest improvement in the state-of-the-art in the *'Supervised, unrestricted, label-free outside data'* protocol[7]. Thus, we conclude that LHS features are competitive in the low complexity domains and are complementary to the high com-

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

Supervised, unrestricted, label free outside data			
#Gauss. (k)	Dimension org. (d_0)	proj. (d)	ROC-EER 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) [14] dataset for different parameter settings.

Methods with similar complexity	
Method	Accuracy
LBP + ITML [51]	85.1 ± 0.6
LBP + PLDA [52]	87.3 ± 0.6
LHS + JML (proposed)	**88.2 ± 0.2**
Methods with multiple feats/higher complexity	
Method	Accuracy
comb. LDML-MkNN [32]	87.5 ± 0.4
comb. PLDA [52]	90.1 ± 0.5
SIFT-FV [34]	93.0 ± 1.1
High dim LBP [53]	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 [53]	95.2 ± 1.1
Deep learning [54, 55]	**97.4 ± 0.3**

Table 4: Comparison with existing works on the Labeled Faces in the Wild (LFW) [14] dataset–unrestricted and supervised setting.

Method	Space dim.	Space reduction	Time ms	Time 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 [34], *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. 4.8

plexity features for supervised face verification on LFW. In the next section we discuss their time and space benefit over the high complexity features.

4.8. Time and space complexity of LHS

The poposed LHS features are very compact and efficient to compute. Compared to one of the state-of-the-art systems for face verification [34] 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) [34].

The best performing LHS features are 10752 dimensional and take 22 ms to compute compared to 67584 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 unoptimized 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.

5. Conclusions

We have presented a model that captures higher-order statistics of small local neighbourhoods 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. Further 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 achives best results for low complexity

features and is complementary to the high dimension 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.

Acknowledgements

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

References

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

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

[3] 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

[4] 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, 4

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

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

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

[8] X. Tan, B. Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions, TIP 19 (6) (2010) 1635–1650. 1, 2, 3, 4

[9] 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, 2, 5, 7, 8

[10] K. Valkealahti, E. Oja, Reduced multidimensional co-occurence histograms in texture classification, PAMI 20 (1) (1998) 90–94. 1, 5

[11] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover Publications, New York, 1966. 1, 5

[12] B. Caputo, E. Hayman, P. Mallikarjuna, Class-specific material categorisation, in: ICCV, 2005. 1, 5, 7, 8

[13] M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba, Coding facial expressions with gabor wavelets, in: AFGR, 1998. 1, 6

[14] 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). 1, 6, 8, 9

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

[16] L. Liu, P. Fieguth, G. Kuang, Compressed sensing for robust texture classification, in: ACCV, 2010. 2

[17] S. Lazebnik, C. Schmid, J. Ponce, A sparse texture representation using local affine regions, PAMI 27 (2005) 1265–1278. 2

[18] 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. 2

[19] M. Varma, A. Zisserman, A statistical approach to texture classification from single images, IJCV 62 (2005) 61–81. 2

[20] M. Croiser, L. D. Griffin, Using basic image features for texture classification, IJCV 88 (2010) 447–460. 2

[21] E. Hayman, B. Caputo, M. Fritz, J.-O. Eklundh, On the significance of real world conditions for material classification, in: ECCV, 2004. 2

[22] Y. Xu, H. Ji, C. Fermuller, View point invariant texture description using fractal analysis, IJCV 83 (2009) 85–100. 2

[23] Y. Xu, X. Yang, H. Ling, H. Ji, A new texture descriptor using multifractal analysis in multi-orientation wavelet pyramid, in: CVPR, 2010. 2

[24] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: CVPR, 2014. 2, 5, 6, 8

[25] D. Lowe, Distinctive image features form scale-invariant keypoints, IJCV 60 (2) (2004) 91–110. 2, 5

[26] T. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: NIPS, 1999. 2, 3, 5

[27] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition, arXiv:1310.1531. 2

[28] C. M. Bishop, Pattern recognition and machine learning, Springer, 2006. 3

[29] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher kernel for large-scale image classification, in: ECCV, 2010. 3

[30] A. Vedaldi, A. Zisserman, Efficient additive kernels using explicit feature maps, in: CVPR, 2010. 3, 5

[31] A. Bellet, A. Habrard, M. Sebban, A survey on metric learning for feature vectors and structured data, arXiv.org. 4

[32] M. Guillaumin, J. Verbeek, C. Schmid, Is that you? Metric learning approaches for face identification, in: ICCV, 2009. 4, 9

[33] A. Mignon, F. Jurie, PCCA: A new approach for distance learning from sparse pairwise constraints, in: CVPR, 2012. 4

[34] K. Simonyan, O. M. Parkhi, A. Vedaldi, A. Zisserman, Fisher vector faces in the wild, in: BMVC, 2013. 4, 5, 8, 9

[35] D. Chen, X. Cao, L. Wang, F. Wen, J. Sun, Bayesian face revisited: A joint formulation, in: ECCV, 2012. 4

[36] 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. 5, 8

[37] C. Shan, S. Gong, P. W. McOwan, Facial expression recognition based on local binary patterns: A comprehensive study, IVC 27 (2009) 803–816. 5, 8

[38] J. Sanchez, F. Perronnin, T. Mensink, J. Verbeek, Image classification with the fisher vector: Theory and practice, IJCV 105 (3) (2013) 222–245. 5

[39] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, http://www.vlfeat.org/ (2008). 5, 8

[40] 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. 6

[41] L. Wolf, T. Hassner, Y. Taigman, Similarity scores based on background samples, in: ACCV, 2009. 6

[42] 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. 8

[43] 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. 8

[44] T. Ojala, K. Valkealahti, E. Oja, M. Pietikäinen, Texture discrimination with multidimensional distributions of signed gray-level differences, PR 34 (3) (2001) 727–739. 8

[45] G. Guo, C. R. Dyer, Simultaneous feature selection and classifier training via linear programming: A case study for face expression recognition, in: CVPR, 2003. 8

[46] M. J. Lyons, J. Budynek, S. Akamatsu, Automatic classification of single facial images, PAMI 21 (12) (1999) 1357–1362. 8

[47] R. S. Javier, V. Rodrigo, C. Mauricio, Recognition of faces in unconstrained environments: a comparative study, EURASIP Journal on Advances in Signal Processing. 7, 8

[48] H. J. Seo, P. Milanfar, Face verification using the LARK rep-

resentation, Information Forensics and Security, IEEE Transactions on 6 (4) (2011) 1275–1286. 7, 8
[49] S. U. Hussain, T. Napoléon, F. Jurie, et al., Face recognition using local quantized patterns, in: BMVC, 2012. 7, 8
[50] D. Yi, Z. Lei, S. Z. Li, Towards pose robust face recognition, in: CVPR, IEEE, 2013. 7, 8
[51] Y. Taigman, L. Wolf, T. Hassner, Multiple one-shots for utilizing class label information, in: BMVC, 2009. 9
[52] P. Li, Y. Fu, U. Mohammed, J. H. Elder, S. J. Prince, Probabilistic models for inference about identity, PAMI 34 (1) (2012) 144–157. 9
[53] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification, in: CVPR, 2013. 9
[54] Y. Sun, X. Wang, X. Tang, Deep learning face representation from predicting 10,000 classes, in: CVPR, IEEE, 2014. 9
[55] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: CVPR, IEEE, 2014. 9