

# Face Recognition using Local Quantized Patterns

Sibt ul Hussain  
Sibt.ul.Hussain@gmail.com

Thibault Napoléon  
Thibault.Napoleon@unicaen.fr

Frédéric Jurie  
Frederic.Jurie@unicaen.fr

GREYC — CNRS UMR 6072,  
University of Caen Basse-Normandie,  
Caen, France

---

## Abstract

This paper proposes a novel face representation based on Local Quantized Patterns (LQP). LQP is a generalization of local pattern features that makes use of vector quantization and lookup table to let local pattern features have many more pixels and/or quantization levels without sacrificing simplicity and computational efficiency. Our new LQP face representation not only outperforms any other representation on challenging face datasets but performs equally well in the intensity space and orientation space (obtained by applying gradient or Gabor Filters) and hence is intrinsically robust to illumination variations. Extensive experiments on two challenging face recognition datasets (FERET [1] and LFW [2]) show that this representation gives state-of-the-art performance (improving the earlier state-of-the-art by around 3%) without requiring neither a metric learning stage nor a costly labelled training dataset, having the comparison of two faces being made by simply computing the Cosine similarity between their LQP representations in a projected space.

## 1 Introduction

Face recognition is an important and popular visual recognition problem that attracts a lot of researchers interest both due to its challenging nature and due to its diverse set of applications. Although several recent commercial systems now use face recognition as a fundamental component, the performance of these systems in uncontrolled environments remains unsatisfactory and therefore motivates for further research in the field.

Advances in the visual representation have been major source of progress in the field of face recognition. Currently, local pattern features (such as Local Binary Patterns (LBP) [3], Local Ternary Patterns (LTP) [4], etc.) are being used quite successfully [5, 6, 7, 8, 9, 10] for face representation because of their ability to robustly encode relevant facial traits as well as their computational simplicity. They have been successfully used in both pixel intensity space [3, 4] and orientation space [7, 8, 10], where these patterns are applied over gradient or Gabor filtered images. Local pattern features harness the local information by finding a qualitative coding for each image pixel and counting the number of occurrences of each possible code over suitable image regions – e.g. over a rectangular grid. Although these local pattern features give good performance, they have limited spatial support as increasing the size of local-neighbourhood increases the histogram dimensions

exponentially. Moreover, these features use hard-wired codings/layouts and are limited to very coarse quantization, mostly binary. These shortcomings limit the overall information encoding capability of these local pattern features and prevent them from leveraging all the available information, thus affect the overall performance of face recognition systems. This paper builds on the recently introduced Local Quantized Patterns (LQP) [8], a generalized and enriched form of the local pattern features, and shows their superiority in characterizing faces. Our face representation is not only fast to compute (no extra computational overhead attached at runtime), but also performs equally well in both the intensity and orientation space and outperforms any existing face representation on different challenging datasets.

In addition to good visual representation, the performance of face recognition systems also depends on the metric used to compare face representations. Most of the recent frameworks learn a dataset specific metric. This results in improving the performance, mainly because it learns the biases (like underlying regularities) of the datasets [22]. In contrast, we show that our face representation can be used within a very simple face-matching algorithm that does not require any pre-labelled dataset, metric-learning algorithm, or any supervised statistical model: once features are computed we use Principal Component Analysis (PCA) to project the features to a low-dimensional space and then perform the spherling of data to have uncorrelated features with the same variance, and finally use angle based Cosine similarity to match faces.

We experimentally validate our face descriptor on two different face recognition tasks: i) *Face verification* (also called authentication), where it has to be decided whether the given pair of images represents the same person or not, for this task we use the popular Labeled Faces in the Wild (LFW) dataset [23]; ii) *Face identification*, where for a given query face image a matching face (if any) is found from a set of images, for this task we use the Face Recognition Technology (FERET) dataset [24].

**Contributions.** This paper introduces a complete framework for face recognition combining (i) a well designed local pattern descriptor with (ii) a simple PCA-based similarity metric to achieve state-of-the-art accuracy rates. In addition to being very simple and efficient, our method has very good generalization capability and not only outperforms any existing unsupervised method but many supervised methods on all the tested datasets.

The rest of paper is organized as follows: Sec. 2 discusses recent related work, Sec. 3 provides the details of the proposed face descriptor and explains the face matching algorithm. Sec. 4 discusses in detail different parameter settings, datasets and provides experimental results. Finally, Sec. 5 concludes the paper with relevant discussion.

## 2 Related Work

Most of the early work [0, 23] used global features implicitly extracted via subspace methods, *e.g.* Eigen and Fisher faces methods [0, 23] project the whole face into a linear subspace to capture the face variations. However, the majority of the best performing methods now use local features for characterizing the facial traits: [3] uses vector quantized local pixels to harness the local information, [15, 16] use a battery of spatially localized Gabor filters in multi-layer framework for face-verification task, [11, 21, 25] use histogram of local pattern features (such as LBP, LTP *etc.*) extracted from intensity-images, [12, 24, 28] use histogram of local binary pattern features extracted from orientation images (obtained via the application of different types of orientation filters, such as Gabor ones). To have better performance on the difficult datasets, recent methods now combine multiple local features such as LBP, LTP, SIFT, Gabor, *etc.* [3, 9, 15, 21, 25, 26], where for combination methods

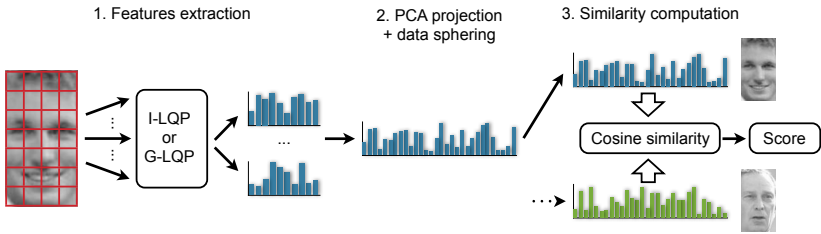


Figure 1: Overview of our face recognition framework.

ranging from simple summing at the decision level to Multiple Kernel Learning (MKL) have been adopted. Although combinations can increase the performance of the final system, it brings extra computational overhead and requires heavy experimental validation to find the best combination model.

Regarding the models used to represent faces, several alternatives have been used, *e.g.* [2, 23] use linear projection models, [8, 6] use linear SVM, [16] linearly combine multiple kernels via MKL, [21] uses kernel LDA. Along with statistical methods, different similarity metrics have also been used to compare faces, those include, among others, distance-based metrics such as Euclidean distance [16, 25] and angle-based Cosine similarity [11, 13, 21, 24]. Recent methods based on these similarity metrics also adopt metric-learning approaches [8, 11, 26] to get further improvement. In addition to holistic methods, researchers have also used component or parts-based methods [8, 6] to cater the higher degree of pose variability.

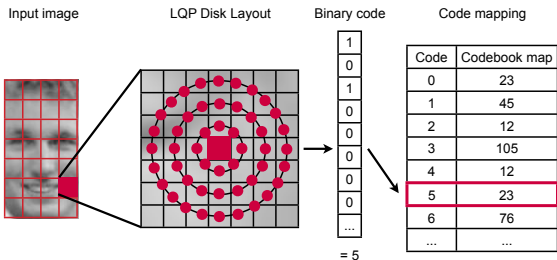
### 3 Face Recognition Framework

Our face recognition framework consists of two stages: first Local Quantized Patterns are used to characterize the facial traits, then Cosine similarity metric is used to match faces in the PCA projected space – *c.f.* Figure 1. The following subsections give relevant details on each of these stages.

#### 3.1 Face Representation using Local Quantized Patterns (LQP)

**Limitations of Local Pattern Features.** Local pattern features (such as LBP [11], LTP [21]) are based on the idea that local neighbourhoods contains a lot of discriminative information and effective visual descriptors can be built by coarsely quantizing the appearance of local neighbourhoods and histogramming the resulting codes. Despite their encoding efficiency and simplicity, these local pattern features remain very local and hence somewhat myopic, where increasing the neighbourhood size by including more pixels or increasing the circle diameter increases the histogram size (number of codes) exponentially, *e.g.* increasing the number of surrounding pixels from 8 to 16 in LBP increases the number of codes from 256 to 65,536 ( $= 2^{16}$ ). Even considering uniform codes (codes containing at most one 0-1 transition [11]) over circles, increase the histogram size quadratically with the increase in circle diameter. Furthermore, in case of large neighbourhood size it is unclear whether to use uniform codes or not because the number of uniform codes (relative to nonuniform ones) becomes infinitesimally small as the number of codes increases. Finally, these local patterns use hard-wired codings and fixed layouts, which are most probably not well adapted to the underlying dataset and/or the application domain. These shortcomings limit the encoding capacity of local patterns and prevent them from harnessing all the information available locally.

Figure 2: Overview of LQP (Disk<sub>7</sub><sup>3\*</sup>) feature computation. LQP samples pixels from a Disk layout around a central pixel and generates a binary/ternary vector and then map the resulting code to nearest codebook word via a pre-built lookup/hash table.



**Local Quantized Patterns (LQP).** LQP [8] is a generalized form of local patterns that uses large local neighbourhoods and/or deeper quantization with domain-adaptive vector quantization to solve many of the above mentioned problems of local patterns. However, to maintain the speed and simplicity of local pattern features and to make the process of vector quantization fast, LQP uses the fact that local patterns binary/ternary codes – produced from the comparisons of surrounding pixel values with the central pixel one – span a discrete space *e.g.* a local pattern that involves 16 ternary pixel comparisons generates  $3^{16} = 43$  million distinct codes, and thus can be easily stored in a lookup table (LUT). LQP builds these tables offline by mapping all the codes to the nearest cluster centers, and thus guarantees fast vector quantization with no extra overhead at runtime. In addition to other advantages, LQP representation also allows for fast codebook learning *i.e.* it counts the number of occurrences of non-zero codes (ignoring those with zero counts) in the dataset and uses this information directly for the fast computation of cluster-centers – *e.g.* doing 10 rounds of tailored K-Means over  $600 \times 10^3$  features of dimension 40 requires only about 28 minutes. LQP features have already been shown to outperform other local pattern features in visual object detection and texture classification tasks [8].

Here we use and tailor these expressive LQP features for the face representation. Precisely, we use the Disk layout (*c.f.* Figure 2) to sample pixels from the local neighbourhood<sup>1</sup> and use a tolerance value ( $\tau$ ) to generate a pair of binary codes (as in LTP) and quantize each one using a separately learned codebook. We will be describing the LQP geometry by notation such as Disk<sub>3</sub><sup>3\*</sup>, where the name Disk represents the type of geometry, the subscript indicates the neighbourhood diameter (here 3 pixels) and the superscript indicates the quantization level (here 3\* denotes split ternary coding as in LTP and 2 binary coding as in LBP). Since our experiments show (*c.f.* Sec. 4) that in case of I-LQP (LQP features computed over raw intensity images) using large-size neighbourhoods give better performance, so in addition to Disks with diameters 3 and 5, we also test Disk with a diameter of 7 pixels<sup>2</sup>.

**Gabor LQP (G-LQP).** Gabor filters have been successfully used for face recognition [10, 15, 21, 28] both due to their robustness to variations in the face appearance (caused by ageing, changes in illuminations, *etc.*) and their close relationship with the receptive fields of simple cells in the mammalian visual cortex [15, 16]. They have been used as a complementary feature set to local pattern features [21] as well as the pre-processing stage of local pattern features [10, 28] where local pattern features are applied on the Gabor filtered images. However, local pattern features computed over a Gabor-filtered image suffer from the same shortcomings as when they are computed on an intensity image. In Gabor LQP (G-LQP), as the name suggests, LQP features are computed from Gabor filtered images obtained by convolving the image with multi-scale multi-orientation Gabor kernels – overall we use 40 different Gabor kernels that span 5 different scales and 8 different orientations

<sup>1</sup>Other layouts such as HVDA, HV, *etc.* (see [8]) gave much worse performance in our initial face recognition experiments.

<sup>2</sup>For this extreme case scenario, we replace LUT with an efficient hashing table and use each distinct code as a hashing key.

over the range 0 to  $2\pi$ . However in contrast to all other methods [10, 18] that compute local pattern features (due to dimensionality problem) independently over different Gabor images, the flexibility of LQP allows it to capture the patterns co-occurrence statistics over neighbouring scales and orientations by concatenating the computed codes from the neighbouring scales and orientations at the local pattern level. This leads to significant improvement in the performance as shown in the experiments – *c.f.* Sec. 4.

**Implementation details.** To learn visual codebook we use weighted K-Means [9] with L2 as inter-vector distance over input patterns represented as binary or ternary vectors. By default we consider only those codes that occur more than a threshold (10 times) in the training dataset, this helps to discard outliers. We run K-Means with ten different random initializations and take the codebook with the least K-Means quantization error over the training set. Once the codebook is learnt, a lookup table is built offline (where possible) by mapping all the codes with both zeros and nonzeros counts to their closest center. To obtain the face descriptor, we overlay the image with a grid of  $10 \times 10$  pixel cells and build cell-level histograms using bilinear interpolation to achieve robustness against small spatial deformations. We then normalize all these cell-level histograms using L1-Sqrt normalization *i.e.* each histogram is normalized to sum one and then square-rooted, and concatenate them to form the final face descriptor.

## 3.2 Comparing Faces using the Cosine Similarity Metric

For comparing face images we use Cosine similarity in a reduced feature space. In the following we provide details on the complete process.

**Principal Component Analysis (PCA)-based reduction.** Our LQP features are of high dimensionality, *e.g.* a  $\text{Disk}_5^{3*}$  descriptor with 150 words codebook computed on  $80 \times 150$  pixel image has a dimensionality of 36,000 ( $= 150 \times 2 \times 8 \times 15$ ). Although they outperform the existing features on the tested datasets (*c.f.* Sec. 4), we still believe like other authors [8, 10, 13, 24] that strong correlations exist between the facial features and that most of the dimensions carry redundant information. Moreover, using high dimensional features makes the descriptor matching process slow, and always includes the risk of over-fitting. So in order to obtain a more discriminant, low-dimensional and uncorrelated facial representation, we use PCA to project the features to a low-dimensional space, where the number of components to retain is decided using a validation set.

**Data Sphering.** The features projected along the principal components are less correlated (as they have a diagonal Covariance matrix) and can be considered as principal facial traits. Nevertheless, they still have varying degrees of variance and PCA strongly favours the dimensions with high variance by weighting more heavily the components corresponding to the large eigenvalues. This scenario might be more acceptable in some other applications, but in the case of face recognition the components having high variance might not necessarily contain the discriminant information, because most of the variance comes from changes in illumination, expression, pose, *etc.* Instead, we believe that discriminant information is equally distributed along all the components. So, like [8, 10, 13, 24], we perform the sphering of data and divide all the principal components by square-roots of their corresponding eigenvalues to have the projected features with the same variance. Performing this normalization reduces the influence of leading principal components by down-weighting their contributions<sup>3</sup> while at the same time increases the influence of trailing components (those with

<sup>3</sup>Empirically, removing the top five components does not have any significant impact on the accuracy.

small eigenvalues) by up-weighting their contributions, and as expected, leads to significant improvement in the performance – *c.f.* Sec. 4.

**Cosine similarity.** Unlike conventional approaches [10, 15, 16, 27] that use a distance-based similarity metric such as Euclidean, and as proposed by [13], by default we use angle-based Cosine Similarity (CS) (*i.e.*  $CS(d_1, d_2) = (d_1^T \cdot d_2) / (||d_1|| \cdot ||d_2||)$ ) metric to compare faces in the normalized projected space. Although we also tested other similarity metrics including the Pearson Correlation Coefficient (PCCS) [13], Chi-squared,  $L_2$ , *etc.*, however in all our experiments CS and PCCS turned out to be the best, both give almost identical and significantly better performance than other tested metrics. Since PCCS turns out to be relatively slow to compute so by default we use CS. We also observed that in the case of other similarity metrics such as Chi-squared and  $L_2$ , it is necessary to perform the normalization of projected descriptors (similar to CS) before doing the comparison to achieve good results.

## 4 Experiments and Discussion

### 4.1 Face Identification on the FERET Dataset

For the problem of face verification, we use the well known Face Recognition Technology (FERET) dataset [24]. FERET contains images of 1,196 different individuals with up to 5 images of each individual captured under different lighting conditions, with non-neutral expressions and over the period of three years. The complete dataset is partitioned into two disjoint sets: gallery and probe. Gallery has labelled images and is used only for training, while the probe images are used for testing purposes. The probe set is further subdivided into four categories: (i) *Fb* images, which are similar to the ones found in gallery with some small variations in expressions; (ii) *Fc* images, which are recorded with different camera and under changing lighting conditions; (iii) *Dup-I* images, taken within the period of thirty-four months; (iv) *Dup-II* images, taken at least one and a half year apart.

**Parameters.** All the images in both the gallery and the probe set are histogram equalized and cropped to have dimensions of  $90 \times 150$  pixels; no other preprocessing is done. We use both the original and flipped versions of images. For a given test image from probe set, we find its closest gallery neighbour and use its label information during the computation of the mean accuracy. Chi-squared distance metric (applied on the unnormalized histograms) is used in the case of original features and Cosine similarity for the projected features. For all experiments, the total number of PCA components has been fixed to 900. In the case of I-LQP, by default we use a 150 words codebook based  $\text{Disk}_7^{3*}$  descriptors with a tolerance value of zero, *i.e.*  $\tau = 0$ . Using a relatively small neighbourhood reduces the performance, *e.g.* reducing the Disk diameter from 7 to 5 reduces the mean accuracy by 2.37%, and so does using higher tolerance values (specially for the *Fc* subset due to weak tolerance to varying lighting conditions), *e.g.* using  $\tau = 5$  gray-level values instead of  $\tau = 0$  reduces the mean accuracy by 0.57% (where 1.55% drop is recorded on *Fc*). Increasing the codebook size to 200 does not lead to any significant improvement, whereas codebooks with more or less number of words affect the performance adversely (similar observations were made for LFW dataset – *c.f.* Figure 3). For G-LQP features we use the same parameter settings except the neighbourhood size and obtain the final matching score by adding the responses from all the filters at the decision-level. By default, we use  $\text{Disk}_3^{3*}$  compound G-LQP features that record co-occurrence statistics over two neighbouring scales and orientations channels, precisely for each filter compound G-LQP uses 150 words codebooks to quantize 40-D vectors obtained from concatenating five (filter itself plus 4 neighbours) 8-bit binary patterns. Compared to compound G-LQP, G-LQP ( $\text{Disk}_5^{3*}$  and  $\text{Disk}_7^{3*}$ ) features computed independently

Methods	Fb	Fc	Dup-I	Dup-II	Mean	Comments
1 HOG [14]	90.0	74.0	54.0	46.6	66.2	
2 LBP [14]	93.0	51.0	61.0	50.0	63.8	
3 LGBPHS [23]	94.0	97.0	68.0	53.0	78.0	Gabor+LBP
4 LGBPWP* [14]	98.1	98.9	83.8	81.6	90.6	Gabor+LBP+WPCA
5 POEM* [24]	99.6	99.5	88.8	85.0	93.2	Gradient+LBP+WPCA+R.Filtering
6 Tan&Triggs [20]	98.0	98.0	90.0	85.0	92.8	Gabor+LBP+DoG Filtering+supervised
7 I-LQP	99.2	69.6	65.8	48.3	70.7	Computed on intensity images
8 I-LQP*	99.8	94.3	85.5	78.6	89.6	
9 G-LQP	99.5	99.5	81.2	79.9	90.0	Computed on Gabor-filtered images
10 G-LQP*	<b>99.9</b>	<b>100.0</b>	<b>93.2</b>	<b>91.0</b>	<b>96.0</b>	

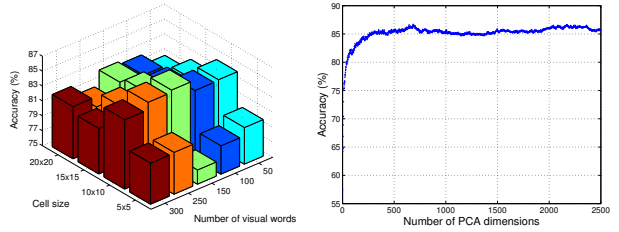
Table 1: Comparative results on FERET dataset. Superscript ‘\*’ is used to differentiate methods using PCA-projected features with Cosine similarity metric from the ones using raw features with Chi-squared distance metric.

over different Gabor-magnitude images reduce the mean accuracy by around 2%. This result emphasizes the fact that indeed recording the co-occurrence statistics among patterns is necessary to achieve good performance.

**Results.** Table 1 gives the accuracy rates of our I-LQP, G-LQP and competing methods on four different probe subsets – we use superscript ‘\*’ to differentiate methods using PCA-projected features with Cosine similarity metric from the ones using raw features with Chi-squared distance metric. I-LQP features significantly outperform the LBP features on three out of four subsets and improve the mean accuracy by 6.9%, where the better performance of I-LQP features can be solely attributed to their rich-encoding ability. Using the Cosine similarity to compare faces in the PCA-projected space also leads to significant improvement in the accuracy, improving the mean accuracy of I-LQP and G-LQP features respectively by 18.9% and 6.0%. Overall, G-LQP features consistently give significantly better performance than the intensity-based I-LQP, especially on the three challenging *Fc* and *Dup-I&II* subsets. This is expected because G-LQP is at the same time more robust against variations due to ageing (*Dup-I&II*) and varying lighting conditions (*Fc*) and contains more discriminant information. In fact, G-LQP\* comes out as the clear winner among all the existing *supervised* and *unsupervised* methods, *e.g.* G-LQP\* gives respectively 2.8% and 3.2% better mean accuracy than earlier state-of-the-art methods of Vu&Caplier (POEM\*) [24] and Tan&Triggs [20], and outperforms them in all four subsets despite not using any complex pre-processing chain like retina or/and DoG filtering – *c.f.* rows 5, 6 and 10. According to the best of our knowledge these are the *best* reported results (both among supervised and unsupervised methods) to date on the FERET dataset.

From the above results we record following observations: (i) local neighbourhoods contain a lot of discriminant information, and LQP features flexible structure enable them to successfully encode this information from large local neighbourhoods; that is otherwise missed by the other local pattern features such as LBP, LTP, *etc.*; (ii) using PCA reduced LQP features with Cosine similarity helps to reduce the computational cost and leads to significant improvements in the accuracy of the system; (iii) Gabor filters (computed on plain images without involving any complex preprocessing chain) based compound G-LQP features remain robust to strong variations introduced by ageing and changing illumination which helps them to achieve state-of-the-art performance on challenging subsets, and hence can be successfully used on datasets captured under similar conditions.

Figure 3: Accuracy of I-LQP features on LFW *View 1* dataset computed (left) using different combinations of cell sizes and number of words in the codebook and trained (right) using different number of PCA components.



## 4.2 Face Verification on LFW Dataset

For the face verification task, we use the popular Labeled Faces in the Wild (LFW) [10] dataset. LFW contains 13,233 face images of 5,749 different individuals of different ethnicity, gender, age, *etc.* It is an extremely challenging dataset and contains face images captured in unconstrained environments with large variations in pose, lighting, clothing, hairstyles, *etc.* LFW dataset is organized into two disjoint sets: ‘View 1’ is used for training and validation (*e.g.* for the configuration of parameters) whereas ‘View 2’ is used for final testing and benchmarking. In our setup, we follow the standard training and evaluation protocol as specified in [10]. We use the aligned version of the faces as provided by Wolf *et al.* [15]<sup>4</sup>.

Although we do compare our results with the supervised methods, still our method is trained using the restricted *unsupervised* framework, *i.e.* during training, our method does not use neither the provided label information nor any outside data from any other source. For the configuration of parameters we use only View 1 dataset. For the evaluation we use the provided 10 random splits of View 2 dataset. We use 4 out of 9 splits for the learning of PCA and the remaining 5 for the configuration of matching threshold (we use the one which gives the best accuracy on validation data) and report the accuracy on the remaining 10th split. We repeat the experiment for 10 times and report the mean accuracy.

**Parameters.** By default, we use the same parameter settings as used for experiments on the FERET dataset, so here we discuss only those parameters which have been different. We resize all the images (of both View 1&2) to  $80 \times 150$  pixel dimensions. In the case of I-LQP features, we use 150 words based  $\text{Disk}_7^{3*}$  descriptors with a tolerance value equal to seven gray-levels ( $\tau = 7$ ) computed using a cell size of  $10 \times 10$ . Using any other combination for cell size and number of words in the codebook gives much worse accuracy (*c.f.* Figure 3-left), so does using smaller tolerance values, since it makes the descriptor less resistant against noise in the near-uniform regions [10] and JPEG artifacts. For G-LQP, we use 150 words based compound  $\text{Disk}_3^{3*}$  – simple  $\text{Disk}_5^{3*}$  and  $\text{Disk}_7^{3*}$  give similar performance – with a tolerance value of zero ( $\tau = 0$ ). For all our experiments we use PCA with 2,000 components, using around 700 components give similar performance on *View 1* (*c.f.* Figure 3-right) but to have a stable system we retain much higher number of components which are still far less than the original 36,000.

**Results.** Table 2 gives our results as well as competing<sup>5</sup> unsupervised and supervised methods on LFW test set (View 2). As observed in the case of FERET dataset: (i) I-LQP and I-LQP\* features significantly outperform other local pattern (LBP and LTP) features, *e.g.* I-LQP and I-LQP\* respectively give 1.8%, 7.0%, 1.6% and 2.9% better accuracy than LBP, LBP\*, LTP and LTP\* features; and (ii) using Cosine similarity over PCA-reduced features gives excellent results. There is no clear winner among I-LQP and G-LQP features as both give identical accuracy *c.f.* rows 9 and 7. However, when used in conjunction with Cosine

<sup>4</sup><http://www.openu.ac.il/home/hassner/data/lfwa/>

<sup>5</sup>Majority of the results are reproduced from the webpage: <http://vis-www.cs.umass.edu/lfw/results.html>



	Methods	Accuracy (%) $\pm s_E$	Methods	Accuracy (%) $\pm s_E$	Comments
Unsupervised	1 SD-MATCHES [10]	64.1 $\pm$ 0.62	11 S.LE+holistic [10]	81.2 $\pm$ 0.53	
	2 GJD-BC-100 [10]	68.5 $\pm$ 0.65	12 DML-eig SIFT [10]	81.3 $\pm$ 0.23	
	3 H-XS-40 [10]	69.5 $\pm$ 0.48	13 Hybrid [10]	84.0 $\pm$ 0.35	
	4 LARK [10]	72.2 $\pm$ 0.49	14 POEM* [10]	84.9 $\pm$ 0.45	
	5 POEM [10]	75.2 $\pm$ 0.73	15 LARK [10]	85.1 $\pm$ 0.59	
	6 POEM [10]	82.7 $\pm$ 0.59	16 LBP + CSML [10]	85.3 $\pm$ 0.52	
	7 G-LQP	75.3 $\pm$ 0.26	17 DML-eig combined [10]	85.7 $\pm$ 0.56	
	8 G-LQP*	82.1 $\pm$ 0.26	18 Combined b/g [10]	86.8 $\pm$ 0.34	10 features+Metric learning+SVM
	9 I-LQP	75.3 $\pm$ 0.80	19 CSML + SVM [10]	88.0 $\pm$ 0.37	6 features+WPCA+CSML+SVM
	10 I-LQP*	86.2 $\pm$ 0.46	20 HTBIF [10]	88.4 $\pm$ 0.58	1000+ Gabor filters+4 distances+SVM

Table 2: Comparative results of our methods with (left) unsupervised and (right) supervised methods on aligned LFW View 2 dataset.

similarity and PCA reduction, I-LQP\* comes out as the clear winner as it gives about 4.1% better accuracy than G-LQP\* while being extremely faster to compute and evaluate. This might be due to the reason that I-LQP works in the intensity space and captures higher level of correlations among the pixels and thus benefits more from the PCA-reduction and data-sphere than G-LQP which works in the orientation-space and therefore does not see the same amount of correlations among pixels – POEM\* which also works in the orientation space, reports similar results (*c.f.* rows 6 and 8).

Overall, our I-LQP\* features give better accuracy than any contemporary unsupervised method and improve the earlier state-of-the-art (POEM\*) by 3.5%. Note that although LQP\* and POEM\* use the same underlying framework to compare faces, still our unsupervised method gives better performance than both unsupervised and supervised (with 1.3% better accuracy – *c.f.* row 6) versions of POEM\*. Consequently, the better performance of our method can be solely attributed to the better discrimination power of LQP features. Cao *et al.* [10] method also resemble ours as they also use locally quantized features with PCA reduction in a SVM framework, but their method’s performance is also about 5% inferior (*c.f.* row 11) to ours. Our *single* feature based unsupervised method also outperforms other supervised methods; precisely, it gives better accuracy rates than 7 out of the 10 supervised ones – *c.f.* rows from 11 to 17. Note that the majority of the supervised methods (*c.f.* rows from 11 to 20) use a battery of features with complex metric learning and/or statistical models and have extremely high computational cost *e.g.* High-Throughput Brain-Inspired Features (HTBIF) based method of [10] uses 1,000+ Gabor filters in a multi-layer architecture with 4 different distance measures and SVMs to achieve good performance – *c.f.* row 20. In comparison to these methods, our method uses a single feature set with a simple similarity metric in unsupervised settings and still gives superior performance than the majority of these supervised methods at a very small computational cost. Actually, according to the best of our knowledge, these are the *best* reported results (both among supervised and unsupervised categories) by a method that uses only a single feature set – nonetheless our method’s performance can still be improved by including the label information and using metric learning with supervised statistical methods.

The main points drawn from our these experiments are mainly the same as observed in the case of FERET dataset except that I-LQP\* features perform better than G-LQP\* from which we can conclude that the choice between whether to compute local patterns in the intensity space or in the orientation space is dataset dependent.

## 5 Conclusions and Discussion

This paper has presented a complete framework for face recognition including a rich visual descriptor for facial representation. We have validated the new representation through

extensive experiments on two challenging datasets, and shown that the proposed representation outperforms any existing face representation. In fact combining this representation with a very generic (not adapted to any dataset) face-matching framework – that neither requires any labelled dataset nor any complex similarity metric-learning algorithm – leads to state-of-the-art performance on challenging real-world datasets. Overall, our main findings are as follows. (i) Carefully adapted expressive features are pivotal to the good performance of the system and can convert a very simple framework to a competitive one. (ii) Although local patterns (such as LBP or LTP) exhibit good performance, they are not fully harnessing the information available locally and thus open the door for more expressive LQP features which due to their flexible structure are able to exploit most of the information available locally, and thus give the better accuracy rates. (iii) The choice between whether to compute local patterns in the intensity space or in the orientation space is dataset dependent.

**Acknowledgments.** This work was partly realized as part of the Quero Program funded by OSEO, French State agency for innovation and by the ANR, grant reference ANR-08-SECU-008-01/SCARFACE.

## References

- [1] T. Ahonen, A. Hadid, and M. Pietikäinen. Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12), 2006.
- [2] P. Belhumeur, P. Hespanha, and J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, august 1997.
- [3] Z. Cao, Q. Yin, X. Tang, and J. Sun. Face recognition with learning-based descriptor. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, USA*, 2010.
- [4] C. Elkan. Using the triangle inequality to accelerate K-Means. In *Proceedings of the 20th International Conference on Machine learning, Washington, USA*, 2003.
- [5] M. Guillaumin, J. Verbeek, and C. Schmid. Is that you? metric learning approaches for face identification. In *Proceedings of the 12th IEEE International Conference on Computer Vision, Kyoto, Japan*, 2009.
- [6] B. Heisele, P. Ho, J. Wu, and T. Poggio. Face recognition: component-based versus global approaches. *Computer Vision and Image Understanding*, 91(1):6–21, 2003.
- [7] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07–49, University of Massachusetts, Amherst, 2007.
- [8] S. Hussain and B. Triggs. Visual recognition using local quantized patterns. In *Proceedings of the 11th European Conference on Computer Vision, Florence, Italy*, 2012.
- [9] N. Kumar, C. Berg, P. Belhumeur, and S. Nayar. Attribute and simile classifiers for face verification. In *Proceedings of the 12th IEEE International Conference on Computer Vision, Kyoto, Japan*, 2009.
- [10] E. Meyers and L. Wolf. Using biologically inspired features for face processing. *International Journal on Computer Vision*, 76:93–104, 2008.
- [11] H. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Asian Conference on Computer Vision*, 2010.
- [12] H. Nguyen, L. Bai, and L. Shen. Local gabor binary pattern whitened PCA: A novel approach for face recognition from single image per person. *Advances in Biometrics*, pages 269–278, 2009.

- [13] V. Perlibakas. Distance measures for PCA-based face recognition. *Pattern Recognition Letters*, 25(6):711–724, 2004.
- [14] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The FERET database and evaluation procedure for face-recognition algorithms. *Transactions on Image and Vision Computing*, 16(5):295–306, 1998.
- [15] N. Pinto and D. Cox. Beyond simple features: A large-scale feature search approach to unconstrained face recognition. In *Automatic Face and Gesture Recognition*, 2011.
- [16] N. Pinto, J. DiCarlo, and D. Cox. How far can you get with a modern face recognition test set using only simple features? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, USA*, 2009.
- [17] J. Ruiz-del-Solar, R. Verschae, and M. Correa. Recognition of faces in unconstrained environments: A comparative study. *EURASIP Journal on Advances in Signal Processing*, 2009:33, 2009.
- [18] H. Seo and P. Milanfar. Face verification using the lark representation. *Information Forensics and Security*, 1:1–12, 2011.
- [19] Y. Taigman, L. Wolf, and T. Hassner. Multiple one-shots for utilizing class label information. In *Proceedings of the 20th British Machine Vision Conference, London, England*, 2009.
- [20] X. Tan and B. Triggs. Fusing Gabor and LBP Feature Sets for Kernel-Based Face Recognition. In *Analysis and Modelling of Faces and Gestures Workshop*, volume 4778, pages 235–249, 2007.
- [21] X. Tan and B. Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, 2010.
- [22] A. Torralba and A. Efros. Unbiased look at dataset bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, USA*, 2011.
- [23] M. Turk and A. Pentland. Face recognition using eigenfaces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Maui, USA*, 1991.
- [24] S. Vu and A. Caplier. Enhanced patterns of oriented edge magnitudes for efficient face recognition. *IEEE Transactions on Image Processing*, 2012.
- [25] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Asian Conference on Computer Vision*, 2009.
- [26] Y. Ying and P. Li. Distance metric learning with eigenvalue optimization. *Journal of Machine Learning Research (Special Topics on Kernel and Metric Learning)*, pages 1–26, 2012.
- [27] B. Zhang, S. Shan, X. Chen, and W. Gao. Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition. *IEEE Transactions on Image Processing*, 16:57–68, 2007.
- [28] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local Gabor Binary Pattern Histogram Sequence (LGBPHS): A Novel Non-Statistical Model for Face Representation and Recognition. In *Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China*, 2005.