PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

LEARNING DISCRIMINATIVE FEATURES FOR FACE RECOGNITION

DANIEL IGNACIO MATURANA SANGUINETI

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ÁLVARO SOTO

Santiago de Chile, January 2012

PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
SCHOOL OF ENGINEERING

LEARNING DISCRIMINATIVE FEATURES FOR FACE RECOGNITION

DANIEL IGNACIO MATURANA SANGUINETI

Members of the Committee:
ÁLVARO SOTO
DOMINGO MERY
ALEJANDRO JARA
MARCELO GUARINI

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, January 2012

To my family

ACKNOWLEDGEMENTS

I gratefully acknowledge my advisor, Professor Álvaro Soto, for his guidance and support during the development of this thesis.

I am also grateful to Professor Domingo Mery for his advice and help in preparing and presenting the work in this thesis.

The financial support of PUC's School of Engineering, the Department of Computer Science, LACCIR and CONICYT is gratefully acknowledged.

Finally, I would like to thank my family for their support and encouragement throughout my studies.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Many state-of-the-art face recognition algorithms use visual descriptors based on features known as Local Binary Patterns (LBPs). While many variations of LBP exist, so far none of them can automatically adapt to the training data. In this thesis we introduce and analyze two methods that learn discriminative LBP-like descriptors for each facial region in a supervised manner.

The first method represents a set of pixel comparisons as a decision tree and builds them by greedily optimizing an entropy-based criterion. This method, Decision Tree Local Binary Patterns (DTLBP), achieves superior accuracy on standard face recognition datasets compared to LBP and other LBP-like approaches, but at a large cost in size of the resulting descriptors.

The second method, Discriminative Local Binary Patterns (DLBPs) simplifies DTLBP by using a non-hierarchical representation of pixel comparisons. It searches for a discriminative set of comparisons by optimizing a Fisher-like separability criterion with stochastic local search. This leads to descriptors that are more compact than DTLBPs and other LBP-like descriptors yet obtain superior or comparable results on standard face recognition datasets.

Keywords: face recognition; local binary patterns; decision tree; id3; hill climbing; nearest neighbor;

RESUMEN

Muchos algoritmos del estado del arte en reconocimiento de caras emplean descriptores basados en características conocidas come Patrones Locales Binarios (PLB). Aunque existen muchas variaciones de los PLB, hasta el momento ninguno puede adaptarse automáticamente a los datos de entrenamiento. En esta tesis introducimos y analizamos dos métodos supervisados para aprender discriminativos descriptores tipo PLB para cada región facial.

El primer método representa un conjunto de comparaciones entre pixeles como un árbol de decisión que es construído con un método codicioso para maximizar un criterio basado en la entropía. Este método, Patrones Locales Binarios basados en Árboles de Decisión (PLBAD), obtiene resultados superiores a PLB y otros descriptores similares en bases de datos estándar en reconocimiento de caras. Sin embargo, estos resultados tienen un costo relativamente alto en términos del espacio ocupado por los descriptores.

El segundo método, Patrones Locales Binarios Discriminativos (PLBD) es una versión simplificada de PLBAD que usa una representación no jerárquica de las comparaciones entre pixeles. Busca un conjunto discriminativo de comparaciones usando un método de búsqueda local estocástico para optimizar un criterio de separabilidad tipo Fisher. Esto crea descriptores más compactos que PLBAD y otros descriptores tipo PLB que obtienen resultados superiores o comparables en bases de datos de reconocimiento de caras estándar.

Palabras Claves: reconocimiento facial; patrones locales binarios; árbol de decisión; id3; búsqueda local; vecino más cercano

1. INTRODUCTION

1.1. Problem Definition

Our goal is to create efficient and discriminative visual features for automatic face recognition from visible light images.

As a specific task in face recognition, we consider *closed set face identification* (Wechsler, 2006). In this task, given a set of identified template images in a face database (often called *gallery*) and an unidentified *probe* image, the goal is to match the probe to an image of the correct individual in the gallery. It is assumed an image of the individual is present in the gallery.

1.2. Motivation

Face recognition has various advantages over other biometric identification systems, like fingerprint analysis or iris scans, such as non-intrusiveness, user friendliness and the relatively low cost of suitable image acquisition devices. These advantages make a robust face recognition system suitable for a wide range of commercial, law enforcement and defense applications. Some example applications include:

- automatically tag pictures of people in online social networks and personal albums
- access control in buildings using cameras
- query law enforcement databases to find if a suspect has outstanding arrest warrants

While classical face recognition algorithms commonly assume that face images are well aligned and have a similar pose, in practical applications such as the above it is often impossible to guarantee these conditions. Therefore extending face recognition to less constrained face images has become an active area of research.

To this end, face recognition systems using features based on properties of small image regions – often known as local visual descriptors or simply local descriptors – have shown

excellent performance on standard face recognition datasets. Examples include the use Gabor features (W. Zhang, Shan, Gao, Chen, & Zhang, 2005; Zou, Ji, & Nagy, 2007; Xie, Shan, Chen, & Gao, 2008), SURF (Bay, Ess, Tuytelaars, & Gool, 2008; Dreuw, Steingrube, Hanselmann, & Ney, 2009), SIFT (Lowe, 2004; Bicego, Lagorio, Grosso, & Tistarelli, 2006), HOG (Dalal & Triggs, 2005; Albiol, Monzo, Martin, Sastre, & Albiol, 2008), and histograms of Local Binary Patterns (LBPs) (Ojala, Pietikinen, & Harwood, 1996; Ahonen, Hadid, & Pietikainen, 2006).

Creating more discriminative and efficient visual features extends the applicability and practicality of face recognition in real world scenarios such as the mentioned above. Moreover, given the similarities between face recognition and other tasks in computer vision, such as object recognition and pedestrian detection, the visual features are potentially also useful for these areas.

1.3. Contributions

The main contribution of this thesis are two methods to learn adaptive, discriminative and efficient visual features that improve on related visual features by learning from labeled training data. We empirically show the effectiveness of these features on standard face recognition datasets, demonstrating the advantage of a learning-based approach.

1.4. Thesis Outline

This thesis is organized as follows. Chapter 2 presents a brief theoretical background in face recognition methods and outlines the general pipeline used by our face recognition system. In chapter 3 we introduce the first version of our approach, Decision Tree Local Binary Patterns (DTLBPs). This chapter is based on Maturana, Mery, and Soto (2010a). Chapter 4 presents Discriminative Local Binary Patterns (DLBP), a refinement of DTLBP that is more efficient and discriminative. This chapter is based on Maturana, Mery, and Soto (2010b) and Maturana, Mery, and Soto (2011). Having presented our main contributions,

chapter 5 discusses related work. Finally, chapter 6 summarizes the main conclusions and gives some avenues for future research.

2. BACKGROUND

In this chapter we begin by describing two major approaches in appearance-based face recognition from still images: holistic and and local. Among the local methods, we describe Local Binary Patterns (LBP), the starting point for our contributions, in slightly more detail. Then we summarize the face recognition pipeline common to many state-of-the-art local appearance-based methods, including ours. Finally, we introduce the face recognition databases we have chosen as benchmarks for our methods.

2.1. Appearance-based Face Recognition

Face recognition is a vast, interdisciplinary field encompassing various tasks and technologies. An exhaustive review exceeds the scope of this thesis; Wechsler (2006) and Li and Jain (2005) are useful introductions. For the purposes of this thesis we will focus on face recognition from photographic images of visible light. Based on how faces are represented and matched, we can distinguish two major categories in this area: holistic and local.

2.1.1. Holistic approaches

Holistic approaches take the whole face images as inputs in the form of real vectors with a dimensionality equal to the image width times its height. The vectors are subjected to a transformation designed to optimize a criterion that varies with each method, but which usually tries to maximize the differences between images corresponding to different people (i.e. interpersonal dissimilarities or distances) while minimizing the differences between images corresponding to the same person (i.e. intrapersonal dissimilarities or distances). In the majority of holistic approaches this transformation is a projection onto a linear subspace of the input.

The best known holistic algorithm is the "Eigenfaces" algorithm (Turk & Pentland, 1991). This algorithm applies Principal Component Analysis (PCA) (Duda, Hart, & Stork, 2000) to the input images and then projects them onto a subspace spanned by the principal

components. These components, which are eigenvectors of the sample covariance of the images, may be visualized as images. The face-like appearance of these images gives the algorithm its name. PCA finds a subspace that minimizes the squared error of the samples with respect to their projections onto the subspace. While this reduces the data dimensionality (when only the largest components are kept) and may reduce variation in the samples due to illumination and other extraneous factors, it is unsupervised. Therefore it does not distinguish between variation caused by intrapersonal versus interpersonal differences in the images.

To address this shortcoming the "Fisherfaces" algorithm was proposed (Belhumeur, Hespanha, & Kriegman, 1997). The Fisherface approach proposes Linear Discriminant Analysis (LDA) to extract a linear subspace that maximizes a so-called "Fisher criterion" (hence the name), which is the ratio of interpersonal variance to intrapersonal variance in the projected images. A solution may be found analytically by solving an generalized eigenvalue problem, provided the input dimensionality is smaller than the number of classes (or identities). In practice this is rarely the case, so PCA is applied to lower the dimensionality of the data prior to LDA. The use of class labels dramatically increased the accuracy of the algorithm over PCA.

The simplicity and success of holistic algorithms have inspired many other variations. Some more recent work in this area includes learning subspaces that preserve neighborhood relationships between samples (Butman & Goldberger, 2008; He, Yan, Hu, Niyogi, & Zhang, 2005), combining subspaces within a classifier ensemble framework (Wang & Tang, 2006) and the use of random projections combined with sparse coding techniques (Wright, Yang, Ganesh, Sastry, & Ma, 2009).

The main drawback of holistic approaches is that they assume that any given pixel in the image corresponds to the same position in the person's face. Therefore, they are most successful in scenarios where the faces image have the same pose, the same expression and are well aligned. Even small violations of these conditions dramatically reduce performance (Ruiz-del-Solar, Verschae, & Correa, 2009; Zou et al., 2007).

2.1.2. Local approaches

In local approaches, selected regions of the face are represented and matched separately. The method to select regions varies between methods. The most two popular methods are using regions centered on facial features (e.g. eyes, nose, mouth) and using a fixed grid of non-overlapping regions. The first is more computationally expensive and requires reliable facial feature detection, but is more robust in scenarios where images are not aligned or have large pose variations. The second approach is simpler and widely used in scenarios where only moderate misalignment and pose variations are expected.

One of the earliest local approaches is Modular Eigenspaces (Pentland, Moghaddam, & Starner, 1994), which simply applied the idea of Eigenfaces to different facial features, obtaining "Eigeneyes", "Eigennoses", and so on. Another influential early approach was Elastic Bunch Graph Matching (Wiskott, Fellous, Kruger, & Malsburg, 1997), which adopted an appearance representation based on the response of a bank of Gabor filters on the selected keypoints. Gabor filters were selected for possessing locality in the spatial and frequency domains, and for their biological plausibility.

Interest in local appearance-based approaches has increased sharply in the last decade, following a trend in other areas of computer vision where approaches based on statistical properties of local regions, often known as "local descriptors" (Mikolajczyk & Schmid, 2005), have become widespread. In face recognition the descriptors used include Gabor features (W. Zhang et al., 2005; Zou et al., 2007; Xie et al., 2008), SURF (Bay et al., 2008; Dreuw et al., 2009), SIFT (Lowe, 2004; Bicego et al., 2006), HOG (Dalal & Triggs, 2005; Albiol et al., 2008), and histograms of Local Binary Patterns (LBPs) (Ojala et al., 1996; Ahonen et al., 2006). The latter forms the basis of our approach and we will describe it in some more detail below. A review and comparison of local descriptor-based face recognition algorithms may be found in Ruiz-del-Solar et al. (2009) and Zou et al. (2007). As observed in these reviews, local approaches tend to be more robust than holistic approaches against moderate changes in pose and expression, because they are based on

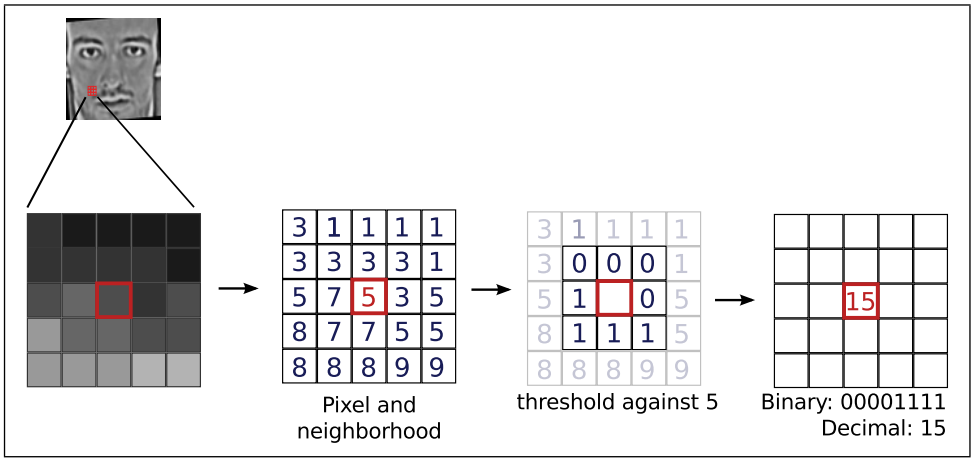

FIGURE 2.1. The LBP operator.

statistical properties of local regions designed to be less sensitive to small changes in the image.

2.1.2.1. Local Binary Patterns

Among the different local descriptors in the literature, histograms of LBPs have become popular for face recognition tasks due to their simplicity, computational efficiency, and robustness to changes in illumination. The success of LBPs has inspired several variations. These include local ternary patterns (Tan & Triggs, 2010), elongated local binary patterns (Liao & Chung, 2007), multi-scale LBPs (Liao, Zhu, Lei, Zhang, & Li, 2007), patch-based LBPs (Wolf, Hassner, & Taigman, 2008), center symmetric LBPs (Heikkilä, Pietikäinen, & Schmid, 2009) and LBPs on Gabor-filtered images (W. Zhang et al., 2005; Xie et al., 2008), to cite a few.

Local binary patterns were originally introduced by Ojala et al. (1996) as a fine scale texture descriptor. In its simplest form, an LBP description of a pixel is created by thresholding the values of a 3×3 neighborhood with respect to its central pixel and interpreting the result as a binary number. An illustration is shown in figure 2.1.

In a more general setting, an LBP operator assigns a decimal number to a pair $(c, \mathbf{n})$,

$$b = \sum_{i=1}^{S} 2^{i-1} I(c, n_i)$$

where c represents a center pixel, $\mathbf{n} = (n_1, \ldots n_S)$ corresponds to a set of pixels sampled from the neighborhood of c according to a given pattern, and

$$I(c, n_i) = \begin{cases} 1 & \text{if } c < n_i \\ 0 & \text{otherwise} \end{cases}$$

This can be seen as assigning a 0 to each neighbor pixel in $\mathbf{n}$ that is smaller than the center pixel c, a 1 to each neighbor larger than c, and interpreting the result as a number in base 2. In this way, for the case of a neighborhood of S pixels, there are 2^S possible LBP values.

We describe how LBPs are used for face recognition in the next section.

2.2. Face Recognition Pipeline

Our face recognition pipeline is similar to the one proposed for LBP in (Ahonen et al., 2006), but we incorporate a more sophisticated illumination normalization step (Tan & Triggs, 2010). Figure (2.2) summarizes its operation, given by the following main steps:

(i) Crop the face region and align the face by mapping the eyes to a canonical location with a similarity transform.

(ii) Normalize illumination with Tan and Triggs' (Tan & Triggs, 2010) Difference of Gaussians (DoG) filter.

(iii) Partition the face image in a grid with equally sized cells, the size of which is a parameter. Let M be the number of grid cells. For each grid cell $m = 1, \ldots, M$, apply a feature extraction operator (such as LBPs or our features, which are described in chapters 3 and 4) to each pixel in the grid cell. Aggregate the feature values into the histogram h_m of size D, where D is the number of possible values of the feature, e.g. in LBP with 8 samples it is 256.

Concatenate the M histograms into a single vector $H = (h_1, \cdots, h_M)$, a vector sometimes known as the "spatial histogram".

(iv) Classify a probe face with the identity of the nearest neighbor in the gallery, where the nearest neighbor distance is calculated with the (possibly weighted)

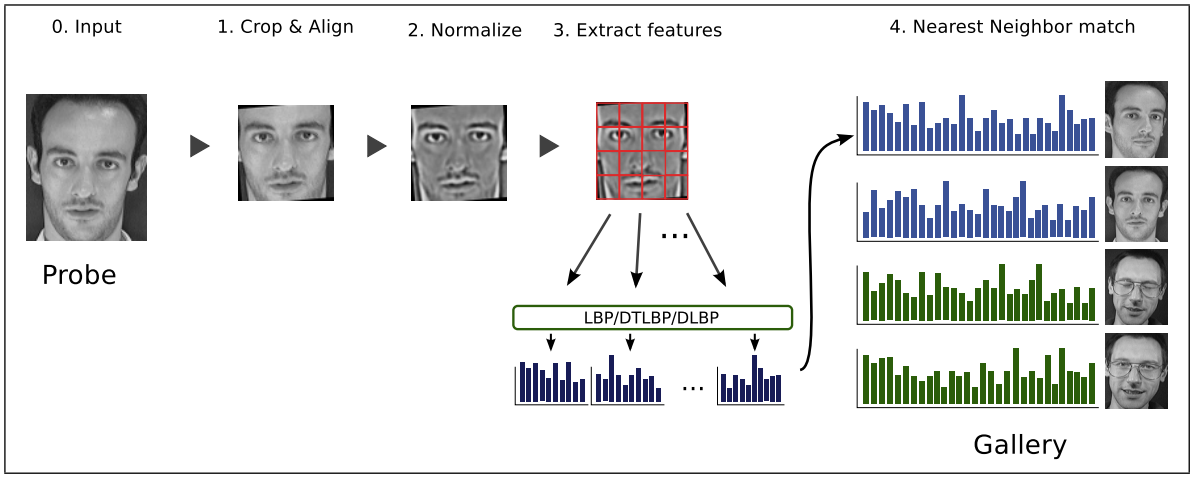

FIGURE 2.2. The face recognition pipeline.

L_1 distance between the histograms of the corresponding face images. In our algorithm, we use one weight for each grid cell. That is, the distance between two spatial histograms $H^1 = (h_1^1, \cdots, h_M^1)$ and $H^2 = (h_1^2, \cdots h_M^2)$ is

$$dist(H^1, H^2) = \sum_{m=1}^{M} w_m \sum_{d=1}^{D} |h_{md}^1 - h_{md}^2| \tag{2.1}$$

where h_{md} is the dth bin of the mth histogram. Unless specified otherwise we use uniform weights. We note that other distance metrics, such as χ^2, histogram intersection and Hellinger distance have also been used. For our descriptors we find these metrics to perform very similarly to each other and to L_1, with L_1 having the advantage of speed.

2.3. Evaluation

We perform experiments on two publicly available datasets, FERET (Phillips, Moon, Rizvi, & Rauss, 2000) and CAS-PEAL-R1 (Gao et al., 2008). These databases have well defined protocols for different face recognition tasks, which eases comparison among different approaches. In particular, both define separate sets of images for the purposes of training and to be used as galleries and probes in the closed set face identification task. It is worth noting that the training dataset is separate from the gallery dataset.

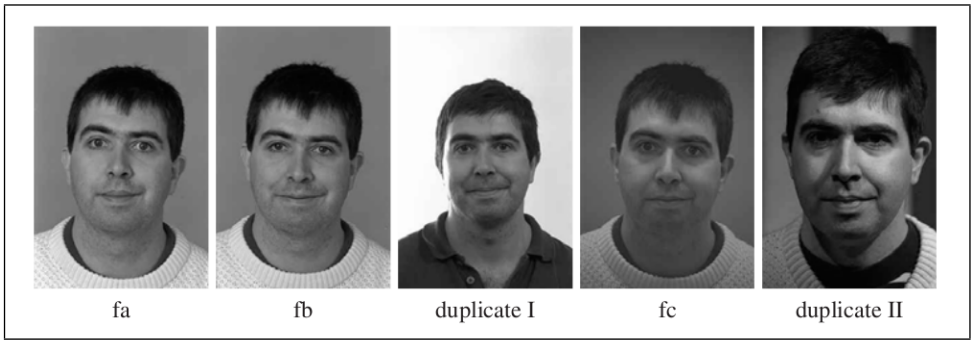

FIGURE 2.3. Images from the FERET database. Montage from Li and Jain (2005).

2.4. FERET

The Facial Recognition Technology (FERET) (Phillips et al., 2000) database was collected as prat of the FERET program, sponsored by the US Department of Defense Counterdrug Techonology Development Program. It was designed to evaluate face recognition algorithms in a relatively controlled setting, with law enforcement and security applications in mind. The dataset images vary in lighting, expression and date of acquisitition.

The FERET database has various sets of images containing different forms of image variation. In this paper we use the most common setup for the closed set face identification test:

Training set: We use 762 images of 429 subjects from FERET training CD. We use the list of training images from the CSU Face Identification Evaluation System package (Bolme, Beveridge, Teixeira, & Draper, 2003), which seems to be a *de facto* standard.

Gallery: The standard frontal pose *fa* gallery (1196 images of 1196 subjects).

Probe sets: The four most commonly used probe sets:

- *fb* (1195 images of 1195 subjects), which has moderate changes facial expression.
- *fc* (194 subjects of 194 subjects), which varies camera and lighting.
- *dup1* (722 images of 243 subjects), containing images taken, on average, 251 days later than the *fa* image.

- *dup2* (234 images of 75 subjects), containing images taken, on average, 627 days later than the *fb* image.

In practice, the illumination and expression also varies between the *dup1* and *dup2* images and their *fa* counterparts.

Figure 2.3 shows example FERET images from each set.

2.5. CAS-PEAL-R1

The Chinese Academy of Sciences Pose, Expression, Accessories and Lighting (CAS-PEAL) (Gao et al., 2008) database explores also contains images taken under controlled conditions and contains sets with variations on pose, expression, use of accessories (such as glasses) and lighting. CAS-PEAL-R1 is a publicly available subset.

In this paper we use the most following setup for the closed set face identification test:

Training set: We use the standard training dataset, with 1200 images of 300 subjects.

Gallery: The standard gallery, with 1040 images of 1040 subjects.

Probe sets: We use the four most commonly used probe sets, with self-explanatory names.

- Accesories (2285 images of 438 subjects).
- Lighting (2243 images of 233 subjects).
- Expression (1570 images of 377 subjects).

Figure 2.3 shows example CAS-PEAL-R1 images.

FIGURE 2.4. Images from the CAS-PEAL-R1 database. Montage from Li and Jain (2005).

3. DECISION TREE LOCAL BINARY PATTERNS

As described in chapter 2, many state-of-the-art face recognition algorithms use image descriptors based on features known as Local Binary Patterns (LBPs). While many variations of LBP exist, so far none of them can automatically adapt to the training data. In this chapter we describe a novel generalization of LBP that learns the most discriminative LBP-like features for each facial region in a supervised manner. Since the proposed method is based on Decision Trees, we call it Decision Tree Local Binary Patterns or DTLBPs. We will then present results on standard face recognition datasets that show the superiority of DTLBP with respect of several state-of-the-art feature descriptors regularly used in face recognition applications.

3.1. Local Binary Patterns as Decision Trees

As we recall from section 2.1.2.1, an LBP operator assigns a decimal number to a pair $(c, \mathbf{n})$, where c represents a center pixel and $\mathbf{n} = (n_1, \ldots n_S)$ corresponds to a set of pixels sampled from the neighborhood of c according to a fixed pattern. This pattern is specified a priori without any input from the data itself.

The simple observation behind DTLBP is that the operation of a LBP over a given neighborhood is equivalent to the application of a fixed binary decision tree. In effect, the aforementioned histograms of LBPs may be seen as quantizing each pair $(c, \mathbf{n})$ with a specially constructed binary decision tree, where each possible branch of the tree encodes a particular LBP. The tree has S levels, where all the nodes at a generic level l compare the center pixel c with a given neighbor $n_l \in \mathbf{n}$. In this way, at each level $l - 1$, the decision is such that, if $c < n_l$ the vector is assigned to the left node; otherwise, it is assigned to the right node. Since the tree is complete, at level 0 we have 2^S leaf nodes. Each of these nodes corresponds to one of the 2^S possible LBPs. In fact, seen as a binary number, each LBP encodes the path taken by $(c, \mathbf{n})$ through the tree; for example, in a LBP with $S = 8$, `11111101` corresponds to a $(c, \mathbf{n})$ pair which has taken the left path at level $l = 1$ and taken the right path at all other levels.

The previous equivalence suggests the possibility of using standard decision tree induction algorithms in place of a fixed tree to learn discriminative LBP-like descriptors from training data. We call this approach Decision Tree Local Binary Patterns or DTLBP. As a major advantage, by using training data to learn the structure of the tree, DTLBP can effectively build an adaptive tree, whose main branches are specially tuned to encode discriminative patterns for the relevant target classes. Furthermore, the existence of efficient algorithms to train a decision tree allows DTLBP to explore larger neighborhoods, such that, at the end of the process the resulting structure of the tree and corresponding pixel comparisons at each node provide more discriminative *spatial histograms*.

Figure 3.1 compares the operation of regular LBPs with respect to DTLBPs. After a decision tree is trained, DTLBP assigns to each leaf node a code given by the path or branch that leads to that node in the tree. In this way, for any input pixel c and the corresponding neighborhood $\mathbf{n}$ used to build the tree, the pair $(c, \mathbf{n})$ moves down the tree according to the $c < n_l$ comparisons. Once it reaches a leaf node, the respective code is assigned to the center pixel c (code number 1 in Figure 3.1). As with ordinary LBPs, the DTLBPs obtained for a given image can be used for classification by building histograms. In summary the proposed approach has the following advantages:

- We can obtain adaptive and discriminative LBPs by leveraging well known decision tree construction algorithms, like ID3 (Quinlan, 1986), as well as more recent randomized tree construction algorithms that have been shown to be very effective in computer vision applications (e.g. Moosmann, Nowak, and Jurie (2008)).
- Since we expect different patterns to be discriminative in different face image regions, we can learn a different tree for each region.
- Instead of neighborhood of eight or sixteen pixels as in regular LPBs, we can use a much larger neighborhood and let the tree construction algorithm decide which neighbors are more relevant.
- Apart from the feature extraction step, DTLBP can be used with no modification in any of the many applications where LBP is currently applied.

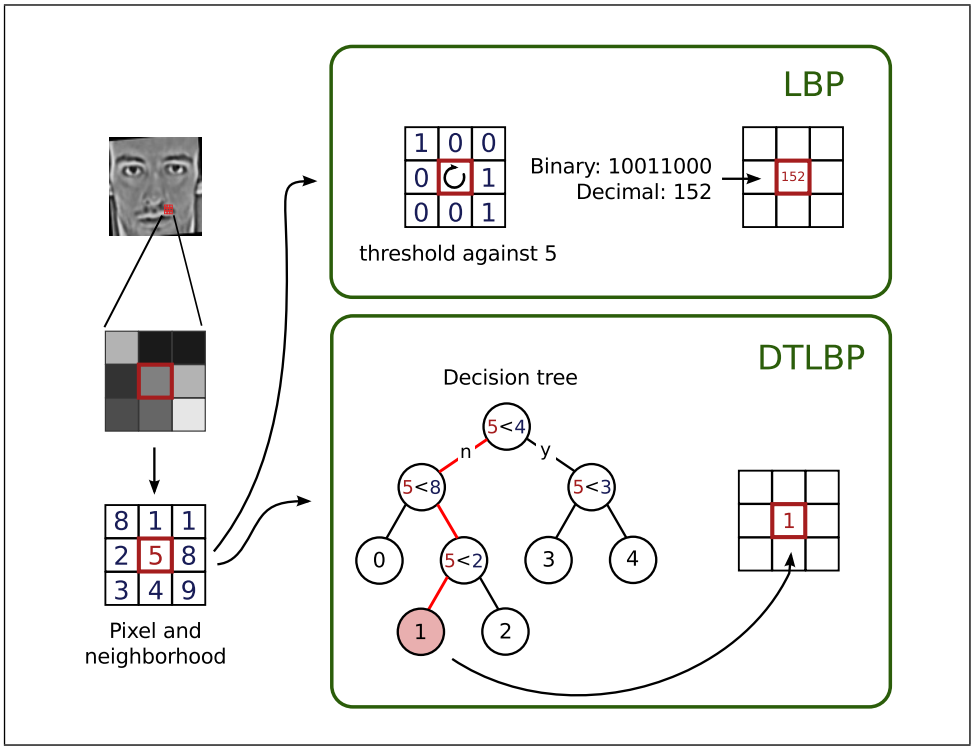

FIGURE 3.1. The LBP operator versus the DTLBP operator.

3.1.1. DTLBP Learning Details

To maximize the adaptivity of our algorithm we learn a tree for each grid cell. The trees are recursively built top-down with a simple algorithm based on Quinlan's classic ID3 method (1986). The algorithm takes as input a "dataset" $\mathcal{X} = \{(c_i, \mathbf{n}_i, y_i)\}_{i=1}^{N}$, a set of tuples where c_i is the value of the center pixel, $\mathbf{n}_i = (n_{i1}, \ldots, n_{is})$ is the vector of values of c_i's neighbors, and y_i is the label of the image from which c_i is taken. These values are taken from the pixels in each grid cell of the images in the training data. The tree construction procedure is summarized in algorithm 1.

Intuitively, `choose_split` chooses a pixel comparison for a node based on how well this comparison separates tuples from different classes. `terminate` yields true if a maximum depth is reached, $|\mathcal{X}|$ is smaller than a size threshold, or there are no informative

Algorithm 1 ID3-based algorithm for DTLBP construction.

`build_tree`$(\mathcal{X}) \equiv$
{Recursively build DTLBP tree}
if `terminate` **then**
  return `LeafNode`
else
  $m \leftarrow$ `choose_split`$(\mathcal{X})$
  `left` $\leftarrow$ `build_tree`$(\{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i \geq n_{im}\})$
  `right` $\leftarrow$ `build_tree`$(\{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i < n_{im}\})$
  return `SplitNode`$(m, \texttt{left}, \texttt{right})$
end if

`choose_split`$(\mathcal{X}) \equiv$
{Choose most informative pixel comparison}
for $d = 1$ to S **do**
  $\mathcal{X}_L \leftarrow \{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i \geq n_{id}\}$
  $\mathcal{X}_R \leftarrow \{(c_i, \mathbf{n}_i, y_i) \in \mathcal{X} \mid c_i < n_{id}\}$
  $\Delta H_d \leftarrow H(\mathcal{X}) - \frac{|\mathcal{X}_L|}{|\mathcal{X}|} H(\mathcal{X}_L) - \frac{|\mathcal{X}_R|}{|\mathcal{X}|} H(\mathcal{X}_R)$
end for
return $\arg\max_d \Delta H_d$

$H(\mathcal{X}) \equiv$
{The class entropy impurity of $\mathcal{X}$. $p(\omega)$ is the fraction of tuples in $\mathcal{X}$ with class label $y_i = \omega$}
return $-\sum_\omega p(\omega) \lg p(\omega)$

pixel comparisons available[1]. The size threshold for $|\mathcal{X}|$ is fixed as 10, and the maximum depth is a parameter.

We define the neighborhood $\mathbf{n}$ used by DTLBP somewhat differently than LBPs. We use a square neighborhood centered around c, and instead of samples taken along a circle, as in regular LBPs, we consider all pixels inside the square as part of the neighborhood (fig. 3.2). All the pixels within this square are considered as potential split candidates. The idea is to let the tree construction algorithm find the most discriminative pixel comparisons.

The main parameters of this algorithm are the size of the neighborhood $\mathbf{n}$ to explore, and the maximum depth of the trees. As shown in Figure 3.2, the first parameter is determined by a radius r. The second parameter, tree depth, determines the size of the resulting

[1]Once a pixel comparison is chosen for a tree node, it provides no information for the descendants of the node. In practice this is used to speed up training.

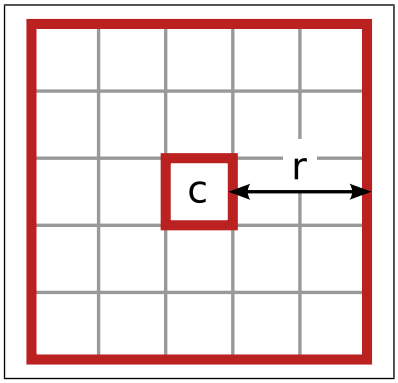

FIGURE 3.2. Pixel neighborhood used in our descriptor. The inner square is the center pixel c, and the neighborhood corresponds to all the pixels enclosed in the larger square. The size of the neighborhood is determined by the radius r.

histograms. Smaller histograms are desirable for space and time efficiency, but as we will show in our experiments, there is a trade-off in accuracy with respect to larger histograms.

Using trees opens up various possibilities. We have explored some extensions to the basic idea, such as using a forest of randomized trees (Shotton, Johnson, & Cipolla, 2008; Moosmann et al., 2008), trees splitting based on a linear combinations of the values of the neighborhood (i.e. nodes split on $\mathbf{n}^T\mathbf{w} < c$, similarly to Bosch, Zisserman, and Muñoz (2007)), or using ternary trees where a middle branch corresponds to pairs for which $|c - n_i| < \epsilon$ for a small ϵ. This last approach can be considered as the tree-based version of the local ternary patterns (Tan & Triggs, 2010). Empirically, we have found that a single binary tree built with an ID3-style algorithm is the best performing solution.

3.2. Experiments

We perform experiments on the FERET (Phillips et al., 2000) and the CAS-PEAL-R1 (Gao et al., 2008) benchmark datasets. First, we examine the effects of the two main parameters of DTLBP: the radius r and the maximum tree depth d. In this case, we measure the accuracy of the algorithm on a subset of FERET. Afterward, we report the accuracy of our algorithm on various standard subsets of FERET and CAS-PEAL-R1 with a selected set of parameters.

In all images we partition the image into an 7×7 grid, as in Ahonen et al. (2006). While in general we have found this partition to provide good results, it is possible that adjusting the grid size to each dataset may yield better results.

For each experiment we show our results along with results from similar works in the recent literature: the original LBP algorithm from (Ahonen et al., 2006); the Local Gabor Binary Pattern (LGBP) algorithm, which applies LBP to Gabor-filtered images; the Local Visual Primitive (LVP) algorithm (Meng, Shan, Chen, & Gao, 2006), which uses K-Means to quantize grayscale patches; the Local Gabor Textons (LGT) algorithm and the Learned Local Gabor Pattern (LLGP) algorithms, which use K-Means to quantize Gabor filtered-images; and the Histogram of Gabor Phase Patterns (HGPP) algorithms, which quantizes Gabor filtered images into histograms that encode not only the magnitude, but also the phase information from the image.

The results are not strictly comparable, since there may be differences in preprocessing and other details, but they provide a meaningful reference. It is worth noting that for each of the algorithm we only show non-weighted variants, since our algorithm does not currently incorporate weights for different facial regions.

3.2.1. Effect of Tree Depth and Neighborhood Size

Figure 3.3 shows the accuracy obtained on FERET *fb* with various combinations of neighborhood sizes and depths. While neighborhood sizes of $r = 1$ and $r = 2$ were also tested, as expected these perform poorly with large trees and are not shown.

We see that larger trees tend to boost performance, however, for some radii there is a point where larger trees decrease accuracy. This suggests that overfitting may be occurring for these radii sizes. We also see that while larger radii tend to perform better, all radii larger than 6 perform similarly. Therefore we set the radius to 7 pixels in the following two experiments.

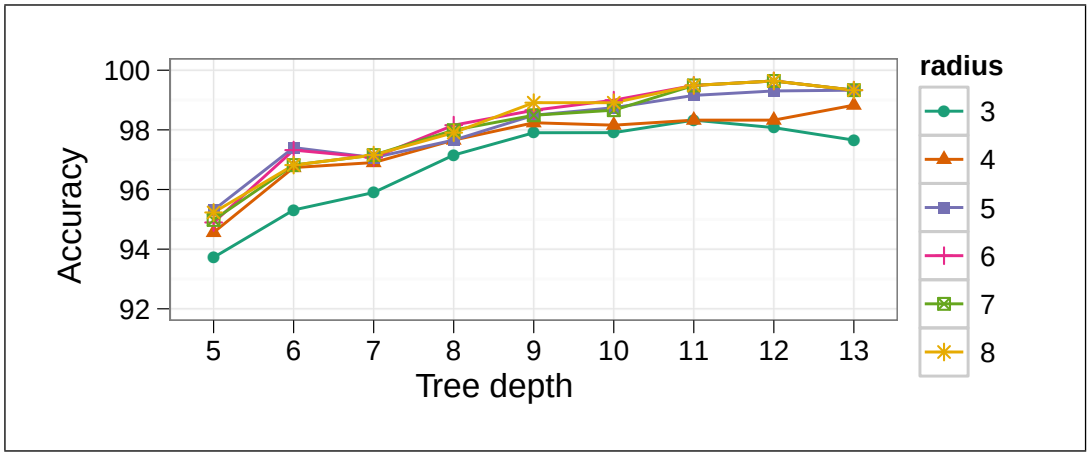

FIGURE 3.3. Effect on accuracy of radius and maximum tree depth in FERET fb.

3.2.2. Results on FERET

For FERET, we use *fa* as gallery and *fb*, *fc*, *dup1* and *dup2* as probe sets. For training, we use the FERET standard training set of 762 training CD images used by the CSU Face Identification Evaluation System package (Bolme et al., 2003).

TABLE 3.1. Accuracy on FERET probe sets. DTLBP^r_d corresponds to a tree of maximum depth d and radius r. TT indicates Tan-Triggs DoG normalization. Accuracies for algorithms other than DTLBP come from the cited papers.

Method	fb	fc	dup1	dup2
LBP (Ahonen et al., 2006)	0.93	0.51	0.61	0.50
LGBP (W. Zhang et al., 2005)	0.94	0.97	0.68	0.53
LVP (Xie, Shan, Chen, Meng, & Gao, 2009)	0.97	0.70	0.66	0.50
LGT (Lei, Li, Chu, & Zhu, 2007)	0.97	0.90	0.71	0.67
HGPP (B. Zhang, Shan, Chen, & Gao, 2007)	0.98	0.99	0.78	0.76
LLGP (Xie et al., 2009)	0.97	0.97	0.75	0.71
DTLBP^7_8, no TT	0.98	0.44	0.63	0.42
DTLBP^7_{10}, no TT	0.98	0.55	0.65	0.47
DTLBP^7_{12}, no TT	**0.99**	0.63	0.67	0.48
DTLBP^7_8	0.98	0.99	0.79	0.78
DTLBP^7_{10}	**0.99**	0.99	0.83	0.78
DTLBP^7_{12}	**0.99**	**1.00**	0.84	0.79
DTLBP^7_{13}	**0.99**	**1.00**	**0.85**	**0.80**

TABLE 3.2. Accuracy on CAS-PEAL-R1 probe sets. DTLBP_d^r corresponds to a tree of maximum depth d and radius r. TT indicates Tan-Triggs DoG normalization. Accuracies for algorithms other than DTLBP come from the cited papers.

Method	Expression	Accessory	Lighting
LGBP (W. Zhang et al., 2005)	0.95	0.87	0.51
LVP (Meng et al., 2006)	0.96	0.86	0.29
HGPP (B. Zhang et al., 2007)	0.96	**0.92**	**0.62**
LLGP (Xie et al., 2009)	0.96	0.90	0.52
DTLBP_8^7, no TT	0.96	0.80	0.20
DTLBP_{10}^7, no TT	**0.99**	0.87	0.23
DTLBP_{12}^7, no TT	**0.99**	0.88	0.25
DTLBP_8^7	0.95	0.89	0.36
DTLBP_{10}^7	0.98	0.91	0.39
DTLBP_{12}^7	0.98	**0.92**	0.40
DTLBP_{13}^7	0.98	**0.92**	0.41

From table 3.1 we can see that our algorithm relies on the Tan-Triggs normalization step to obtain competitive results on the probe sets with heavy illumination variation. When the normalization step is included, our algorithm obtains the best results on all the probe sets. We argue that the DoG filter in the Tan-Triggs normalization plays a similar role to the Gabor filters in the Gabor-based algorithms, but is much more efficient computationally.

3.2.3. Results on CAS-PEAL-R1

Results for this dataset are summarized in table 3.2. In the Expression probe set, which does not have intense illumination variation, DTLBP without illumination normalization obtained the best results, and DTLBP with normalization the second best. The algorithm with normalization obtains the best result, along with HGPP, in the Accessory probe set. On the Lighting dataset, the overall performance of all the algorithms is rather poor. In this case, the best results are given by HGPP, but LGBP and LLGP also obtain good results. All these algorithms use features based on Gabor wavelets, which suggests that Gabor features provide more robustness against the extreme lighting variations in this dataset than the DoG filter.

3.3. Discussion

The results show that DTLBPs are highly discriminative features. Their discriminativity increases as the trees grow, but this has an exponential impact in the computational time and storage cost of using these features. For example, a tree of maximum depth 8 corresponds to a maximum of 256 histogram bins, while a tree with maximum depth 13 corresponds to a maximum of 8192 bins. Since we use $7 \times 7 = 49$ grid cells, the total number of histogram bins in each spatial histogram is at most 401,408 bins. In practice, we find that our C++ implementation is fast enough for many applications – converting an image to a DTLBP spatial histogram and finding its nearest neighbor in a gallery with more than a thousand images takes a couple of seconds. However, the cost in terms of memory and storage becomes an obstacle to the use of larger trees. For example, a gallery of 1196 subjects with 49 grid cells and trees of maximum depth 13 takes about 1.8 GB of storage when stored naively. However, the resulting dataset is very sparse, which can be taken advantage of to compress it. A straightforward solution is keep the most popular bins, and discard the rest or merge them into a single bin. This is analogous to the so-called *uniform patterns* used by traditional LBPs.

3.4. Summary

We have proposed a novel method that uses training data to create discriminative LBP-like descriptors by using decision trees. The algorithm obtains encouraging results on standard datasets, and presents better results that several state-of-the-art alternative solutions. In particular, with respect to a face recognizer based on the widely used LBPs, our approach presents a significant increase in accuracy, demonstrating the advantages of using an adaptive and discriminative set of local binary patterns.

4. DISCRIMINATIVE LOCAL BINARY PATTERNS

Chapter 3 presented Decision Tree Local Binary Patterns (DTLBPs), an adaptive and discriminative variation of Local Binary Patterns (LBPs). Experiments showed DTLBPs provided better classification accuracy than LBPs, but this accuracy came at a cost in the space requirements of the descriptors.

In this chapter we will present a method to learn adaptive and discriminative LBP-like descriptors that are simpler and more efficient than DTLBPs. The method, which we simply call Discriminative Local Binary Patterns (DLBPs) represents the descriptor as a set of pixel comparisons within a neighborhood and heuristically seeks for a set of pixel comparisons so as to maximize a Fisher separability criterion for the resulting histograms. We present tests on standard face recognition datasets that show that this method outperforms DTLBP and other state-of-the-art descriptors.

4.1. Finding Discriminative Local Binary Patterns

As in the previous chapter, let c be a center pixel and $\mathbf{n} = (n_1, \ldots n_S)$ the neighbor pixels. Traditionally, the neighbor pixels are sampled in a circular shape that is parameterized by S and r, the radius in pixels of the circle. Thus, the 3×3 descriptor corresponds to $S = 8$ and $r = 1$. But other configurations are possible. For example, Liao and Chung (2007) proposes the use of elliptical shapes, parameterized by the length of the axes, and Wolf et al. (2008) propose a pattern with two rings. In place of these hand-crafted shapes we propose to learn the best patterns in a supervised fashion. In particular, the set of neighbors $\mathbf{n} = (n_1, \ldots, n_S)$ is not determined by a parametric form but may correspond to arbitrary pixels within a small distance from the center, as seen in figure 4.1. Thus, the space of possible patterns in our method is determined by two parameters: r and S. As in chapter 3, r is the size of the neighborhood, and has a different meaning than in LBP; the neighbors n_i must be within a square neighborhood centered on c, but not necessarily at distance r (see figure 3.2). S, as in LBP, is the number of samples. In general, a larger S results in better classification accuracy, but has a larger cost in computation and storage.

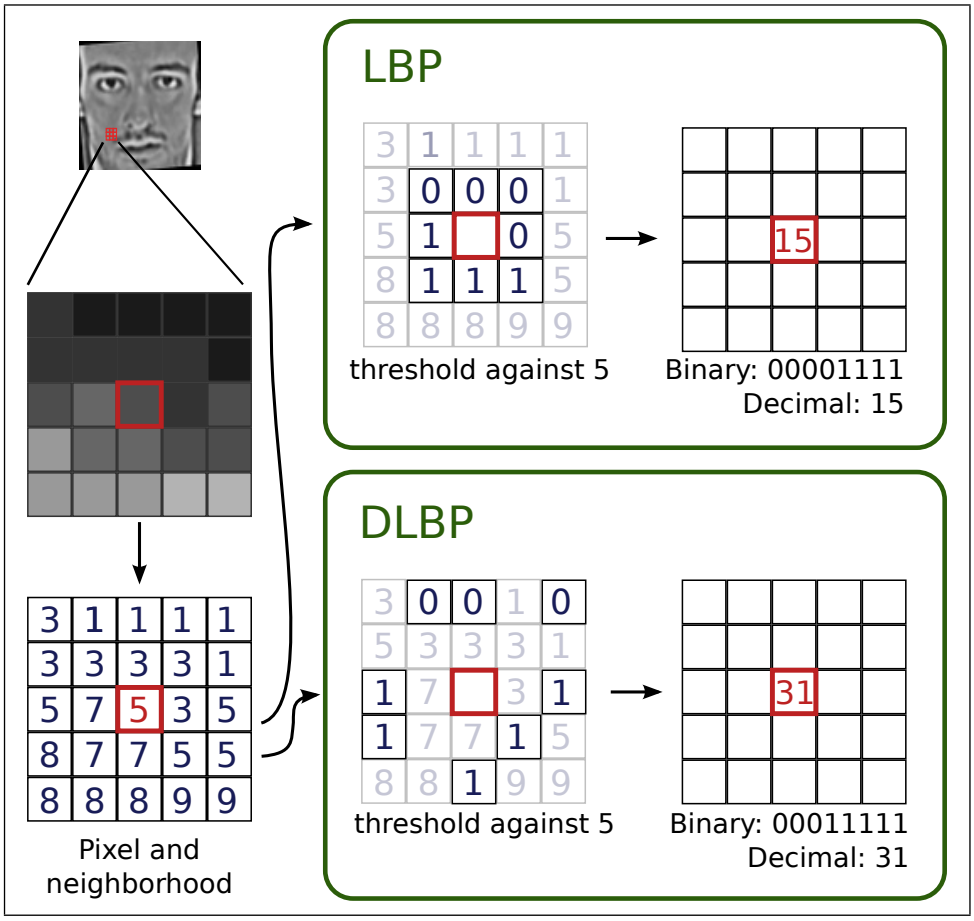

FIGURE 4.1. The LBP operator versus the DLBP operator. In the LBP operator, pixel comparisons are restricted to a predetermined pattern; in DLBP, the pattern is learned discriminatively.

Within the square neighborhood given by r, there are $(2r+1)^2 - 1$ possible pixel comparisons. We wish our DLBP operator to consist of a subset $\mathbf{n}$ of those comparisons of size S that maximizes the discriminativity of the output histograms. To quantify discriminativity we use a Fisher-like class separability criterion:

$$J = \frac{(\mu_w - \mu_b)^2}{\sigma_w^2 + \sigma_b^2} \tag{4.1}$$

where μ_w and μ_b are the mean within-subject and between-subject distances of the histograms induced by the set $\mathbf{n}$, and σ_w^2 and σ_b^2 are the variances of the within-subject and between-subject distances of the histograms induced by the set. This criterion was used by

Zhang et al (2005) to weigh different facial regions. We also tested other critera for discriminativity, such as the multiclass Fisher criterion used in Fisherfaces (Belhumeur et al., 1997), and a margin-based criterion (Gilad-Bachrach, Navot, & Tishby, 2004) but found them to perform slightly worse.

Unfortunately, finding the set $\mathbf{n}$ of comparisons that maximizes J subject to the size constraint of S is an intractable combinatorial optimization problem. Therefore we use a simple iterative heuristic algorithm, stochastic hill climbing, to obtain an approximate solution. Other heuristics, such as simulated annealing or Tabu search could also be used. Informally, the algorithm begins with a random solution (a random set of pixel comparisons, in our case) and iteratively attempts to improve it by making small modifications (swapping a pixel comparison by a different one). The hill climbing procedure is summarized in algorithm 2.

We run this algorithm five times and store the best set obtained during the hill climbing phase along with its associated J^* value.

Since we expect different patterns to be discriminative in different face regions, we learn a new DLBP for each grid cell. One of the side benefits of the optimization scheme is that we may use the J^* obtained for each cell as a weight for the distance calculation in (2.1), assuming that J^* reflects how discriminative is the facial region corresponding to the grid cell.

The optimization process has its own set of parameters, namely the number of hill climbing iterations and the number of tweaks tested at each hill climbing iteration. These are dictated mostly by the available computing resources. We use 60 hill climbing iterations and 30 tweaks tested per iteration. With these parameters, our C++ implementation of the training process takes around 4 hours (in total, for all face regions) on a 1.6GHz laptop with $S = 8$.

Algorithm 2 Hill climbing algorithm for DLBP construction.

```
hillclimb(set) ≡
J* ← −∞
for i ← 1 to hill_climbing_iterations do
   new_set ← copy(set)
   J' ← J(new_set)
   for j ← 1 to tweaks do
      set' ← tweak(copy(set))
      if J(set') ≥ J' then
         J' ← J(set')
         new_set ← set'
      end if
   end for
   set ← new_set
   if J' ≥ J* then
      best ← set
      J* ← J'
   end if
end for
return best
```

J(*set*) ≡
Quantize data into histograms with *set*
Evaluate and return $(\mu_w - \mu_b)^2 / (\sigma_w^2 + \sigma_b^2)$

tweak(*set*) ≡
Select random pixel comparison n_i from *set*
Set n_i to another random pixel within the neighborhood that does not already belong to *set*
return *set*

4.2. Discriminative Local Binary Patterns versus Decision Tree Local Binary Patterns

DLBPs can be seen as a simplification and an improvement to DTLBPs. DTLBPs represents the descriptor as a tree of pixel comparisons, whereas DLBPs simply uses a list. DTLBP descriptors are far more flexible than DLBPs; DLBPs can be seen as a special case of DTLBP where every path from the root of the tree to a leaf has the same set of pixel comparisons. This flexibility corresponds to a larger number of parameters (possible pixels to compare against), which makes it more difficult to find good solutions. In practice, DTLBPs are recursively grown with a ID3-like (Quinlan, 1986) algorithm. This algorithm

has two drawbacks. One is its greediness, which may lead it to find worse solutions than algorithms that perform a more exhaustive search. Another is that the entropy criterion that is optimized is only indirectly related to our objective of creating histograms that maximize interpersonal distances while minimizing intrapersonal distances.

The DLBP method addresses the first shortcoming by using a more exhaustive local search heuristic, that coupled with random restarts leads to a better exploration of the space of possible solutions. The simpler representation also means the space of solutions is smaller and easier to search. While in theory the flexibility of trees could lead to better performing descriptors, in practice we have found it not to be the case, even when combining hill climbing and random restart techniques with ID3 (Maturana et al., 2010b). The second shortcoming is addressed by using optimizing the Fisher criterion instead of entropy gain. The Fisher criterion explicitly captures our objective of making more "separable" histograms that maximize nearest neighbor performance.

4.3. Experiments

As in chapter 3 we perform experiments on the FERET (Phillips et al., 2000) and the CAS-PEAL-R1 (Gao et al., 2008) benchmark databases. We report results with and without weights, where the weights for each region are set as the final J value from (4.1) for the set of each region.

Regarding the parameters, in order to give the algorithm flexibility in the choice of patterns we use a relatively large radius, $r = 7$ [1]. This was the radius used in chapter 3. Figure 4.3 illustrates the effect of radius and neighborhood samples on the accuracy of DLBP on FERET fb. The trend, also seen in other datasets, is that all radii larger than 2 perform comparably. S is varied to evaluate the size-accuracy tradeoff. In all images we partition the image into an 7×7 grid, as originally used in (Ahonen et al., 2006). While in general we have found this partition to provide good results, it is likely that adjusting the grid size to each database may yield better results.

[1] To handle the border pixels, we simply added a black border of width 7 to each image. More sophisticated schemes gave similar or worse results.

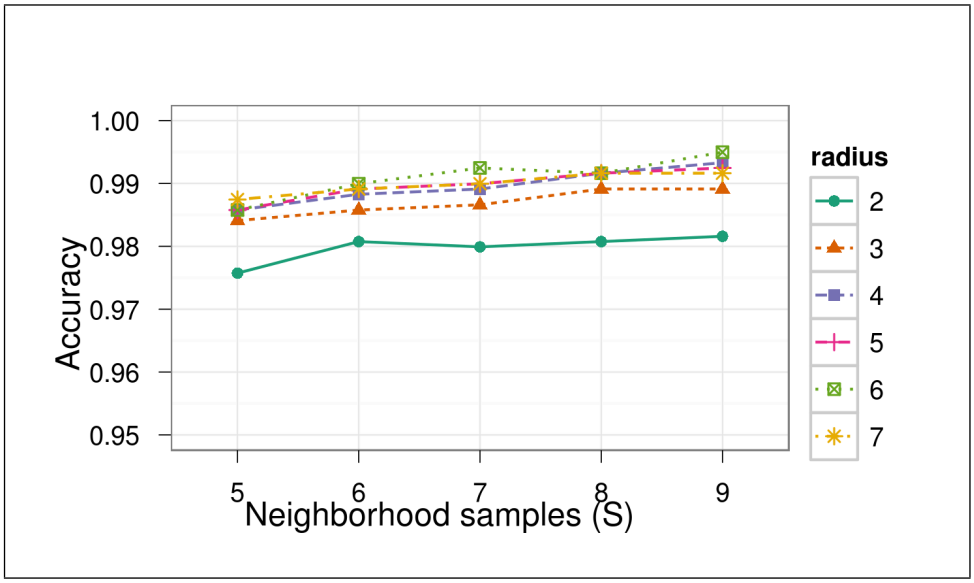

FIGURE 4.2. Effect of neighborhood samples (S) and radius on accuracy of DLBP on FERET fb.

For each experiment we show our results along with the best results from similar works in the recent literature: the original LBP algorithm from Ahonen et al. (2006); the Local Gabor Binary Pattern (LGBP) algorithm (W. Zhang et al., 2005), which applies LBP to Gabor-filtered images; the Local Visual Primitive (LVP) algorithm (Meng et al., 2006), which uses K-Means to quantize grayscale patches; the DTLBP algorithm from chapter 3, which uses decision trees to find discriminative LBPs on grayscale images; Local Gabor Textons (LGT) algorithm (Lei et al., 2007) and the Learned Local Gabor Pattern (LLGP) (Xie et al., 2009) algorithm, which use K-Means to quantize Gabor filtered-images; and the Histogram of Gabor Phase Patterns (HGPP) algorithm (B. Zhang et al., 2007), which quantizes Gabor filtered images into histograms that encode not only the magnitude, but also the phase information from the image. For each algorithm, if a weighting scheme is used, we show the best results with the weighting scheme under the name of the algorithm followed by '-W'. We also show the results of using a purely random set of pixel comparisons as RLBP (for Random LBP) to assess the effects of the supervised optimization. For the LBP algorithm, we give the accuracy obtained by the original authors as well as by our own implementation. The reason is that due to the different preprocessing and border handling

the accuracies differ. In addition, for LBP we add results with a radius of 7, since DLBP uses a radius of that scale, and also add results with the weight given by the same Fisher J value as in DLBP. We also show results with and without Tan-Triggs normalization to show the effect this step has on the results.

The results cited from other papers are not strictly comparable, since there may be differences in preprocessing, and for FERET, the training set used, but they provide a meaningful reference.

4.3.1. Results on FERET

For FERET, we use *fa* as gallery and *fb*, *fc*, *dup1* and *dup2* as probe sets. For training, we use the FERET standard training set of 762 images from the training CD provided by the CSU Face Identification Evaluation System package (Bolme et al., 2003). The results are summarized in tables 4.1 and 4.2.

We can see that our algorithm does well on FERET, specially when normalization is used. Without normalization, DLBP's accuracy suffers on fc, which varies illumination. With Tan-Triggs normalization it obtains the best results on fb, $dup1$ and $dup2$, and comparable to the best on fc.

4.3.2. Results on CAS-PEAL-R1

In CAS-PEAL-R1 we use the standard training and gallery subsets, and we use the Expression, Lighting and Accessory subsets as probes. The results are summarized in tables 4.3 and 4.4.

In this dataset our algorithm also does well. It obtains the best results in the Expression and Accessory datasets, tying with HGPP in the latter. As before, without normalization the performance of DLBP suffers on datasets with illumination variation. On the lighting dataset, the overall performance of all the algorithms is rather poor. In this case, the best performance are given by LGBP, HGPP and LLGP. All these algorithms use features based on Gabor wavelets, which suggests that Gabor features provide robustness against the extreme lighting variations in this dataset.

TABLE 4.1. Accuracy on FERET probe sets. DLBP_d^r corresponds to a set of size d and radius r. "-W" indicates weights. "no TT" indicates no illumination normalization.

	No TT				With TT			
Method	fb	fc	dup1	dup2	fb	fc	dup1	dup2
LBP_8^2	.96	.53	.60	.40	.93	.96	.72	.67
LBP_8^7	.96	.34	.61	.45	.98	.96	.79	.78
RLBP_5^7	.94	.28	.57	.35	.95	.93	.71	.66
RLBP_6^7	.95	.28	.57	.36	.96	.95	.75	.75
RLBP_7^7	.96	.38	.59	.37	.97	.95	.78	.73
RLBP_8^7	.96	.32	.60	.39	.97	.97	.80	.75
RLBP_9^7	.97	.44	.61	.42	.98	.97	.83	.81
DLBP_5^7	.96	.29	.61	.40	.97	.94	.77	.75
DLBP_6^7	.97	.31	.61	.41	.98	.97	.80	.79
DLBP_7^7	.98	.39	.63	.44	.98	.98	.81	.80
DLBP_8^7	.98	.37	.64	.47	.98	**.99**	.84	.82
DLBP_9^7	.98	.40	.66	.48	**.99**	**.99**	.85	.84
LBP-W_8^2	.98	**.54**	.62	.45	.97	.97	.72	.68
LBP-W_8^7	.98	.32	.65	.53	**.99**	.97	.81	.80
RLBP-W_5^7	.97	.27	.59	.41	.97	.93	.72	.69
RLBP-W_6^7	.98	.30	.61	.45	.98	.94	.77	.76
RLBP-W_7^7	.98	.36	.63	.48	**.99**	.97	.79	.74
RLBP-W_8^7	.98	.31	.65	.53	**.99**	.97	.81	.78
RLBP-W_9^7	.98	.38	.65	.54	**.99**	.98	.81	.79
DLBP-W_5^7	.98	.28	.63	.47	.98	.94	.78	.78
DLBP-W_6^7	**.99**	.34	.63	.49	**.99**	.98	.82	.81
DLBP-W_7^7	**.99**	.42	.66	.53	**.99**	.98	.85	**.85**
DLBP-W_8^7	**.99**	.41	.67	.54	**.99**	**.99**	.85	**.85**
DLBP-W_9^7	**.99**	.48	**.68**	**.55**	**.99**	**.99**	**.86**	**.85**

4.3.3. Discussion

The results show that DLBPs are highly discriminative features. However, it seems that without normalization the learning process tends to overfit on datasets with intense illumination variation and performance suffers. It should be noted that the normalization is a computationally inexpensive process; it consists in convolution with a Difference of

Method	fb	fc	dup1	dup2
LBP (Ahonen et al., 2006)	.93	.51	.61	.50
LGBP (Xie et al., 2009)	.94	.97	.68	.53
LVP (Xie et al., 2009)	.97	.70	.66	.50
LGT (Lei et al., 2007)	.97	.90	.71	.67
HGPP (B. Zhang et al., 2007)	.98	.99	.78	.76
LLGP (Xie et al., 2009)	.97	.97	.75	.71
LBP-W (Ahonen et al., 2006)	.97	.79	.66	.64
LGBP-W (Xie et al., 2009)	.98	.97	.74	.71
LVP-W(Xie et al., 2009)	**.99**	.80	.70	.60
HGPP-W (B. Zhang et al., 2007)	.98	.99	.78	.77
LLGP-W (Xie et al., 2009)	**.99**	.99	.80	.78
DTLBP (Maturana et al., 2010a)	**.99**	**1.0**	.84	.80
DLBP-W_9^7 (ours)	**.99**	.99	**.86**	**.85**

TABLE 4.2. Comparison of accuracy with other algorithms on FERET probe sets. Figures from entries with citations come from the respective citation.

Gaussians filter and a couple of equalization steps. This is much faster than convolution with the real and imaginary parts of 40 Gabor filters, as in LGBP, LGT and HGPP.

As expected, the supervised optimization process improves upon the purely random descriptor, though in some cases the difference is relatively small. Thus RLBP could be of interest for unsupervised scenarios.

We highlight the the ability our algorithm to create compact yet discriminative descriptors. Even when using $S = 5$, which yields histograms of size 32, it performs comparably or better than (non-uniform) LBP, of size 256. Our largest and best-performing descriptor ($S = 9$) yields histograms of size 512, which are smaller than those used by Gabor-based approaches that concatenate histograms for each Gabor orientation and scale; for example, LLGP uses 12 Gabor filters and a codebook of $K = 70$ for each, giving histograms of size $12 \times 7 = 840$; LGBP (W. Zhang et al., 2005) uses 40 Gabor images and an LBP of $S = 8$ for each, resulting in histograms of size $256 \times 40 = 10240$. DLBP's histograms are also much smaller than the ones used for DTLBP; the results we show are from a descriptor with histograms of size 2^{13}.

Method	No TT			With TT		
	Exp.	Acc.	Light.	Exp.	Acc.	Light.
LBP_8^2	.95	**.82**	.19	.97	.89	.29
LBP_8^7	.94	.72	.17	.94	.85	.27
$RLBP_5^7$	.90	.65	.14	.91	.81	.24
$RLBP_6^7$	.92	.69	.15	.91	.82	.23
$RLBP_7^7$	.93	.7	.15	.93	.84	.25
$RLBP_8^7$	.94	.75	.16	.95	.86	.27
$RLBP_9^7$	.93	.72	.17	.95	.87	.28
$DLBP_5^7$	.94	.73	.18	.96	.88	.31
$DLBP_6^7$	.95	.77	.20	.97	.90	.35
$DLBP_7^7$	.96	.79	.21	.97	.90	.36
$DLBP_8^7$	**.97**	.81	.22	.98	.91	.39
$DLBP_9^7$	**.97**	**.82**	.23	.98	**.92**	.40
$LBP\text{-}W_8^2$	**.97**	.78	.20	.97	.89	.31
$LBP\text{-}W_8^7$	.94	.71	.18	.93	.86	.26
$RLBP\text{-}W_5^7$	.89	.65	.15	.91	.81	.24
$RLBP\text{-}W_6^7$	.92	.67	.17	.92	.83	.23
$RLBP\text{-}W_7^7$	.94	.69	.16	.93	.85	.24
$RLBP\text{-}W_8^7$	.94	.75	.18	.95	.87	.27
$RLBP\text{-}W_9^7$	.93	.74	.18	.94	.88	.28
$DLBP\text{-}W_5^7$	.95	.72	.20	.96	.87	.31
$DLBP\text{-}W_6^7$	.95	.76	.22	.97	.90	.35
$DLBP\text{-}W_7^7$	.96	.78	.22	.98	.90	.36
$DLBP\text{-}W_8^7$	**.97**	.81	.23	.98	**.92**	.40
$DLBP\text{-}W_9^7$	**.97**	**.82**	**.24**	**.99**	**.92**	**.41**

TABLE 4.3. Accuracy on CAS-PEAL-R1 probe sets. $DLBP_d^r$ corresponds to a set of size d and radius r. "-W" indicates weights. "no TT" indicates no illumination normalization.

4.4. Summary

We have proposed a novel method that uses training data to learn compact and discriminative LBP-like descriptors. The algorithm obtains encouraging results on standard databases, and presents better results that several state-of-the-art alternative solutions. In particular, with respect to a face recognizer based on DTLBP, DLBPs are more accurate and efficient.

Method	Expression	Accessory	Lighting
LGBP (Xie et al., 2009)	.95	.87	.51
LVP (Meng et al., 2006)	.96	.86	.29
HGPP (B. Zhang et al., 2007)	.96	**.92**	.62
LLGP (Xie et al., 2009)	.96	.90	.52
LVP-W (Meng et al., 2006)	.96	.86	.33
HGPP-W (B. Zhang et al., 2007)	.97	**.92**	**.63**
LLGP-W (Xie et al., 2009)	.96	**.92**	.55
DTLBP (Maturana et al., 2010a)	.98	**.92**	.41
DLBP-W_9^7 (ours)	**.99**	**.92**	.41

TABLE 4.4. Comparison of accuracy with other algorithms on CAS-PEAL-R1 probe sets. Figures from entries with citations come from the respective citation.

5. RELATED WORK

Our algorithm can be seen as a way to quantize local image patches using a codebook where each code corresponds to a leaf node in DTLBP or one of the 2^S possible codes in DLBP. This links our algorithm to various other works in vision that use codebooks of image features to represent images, which we describe in section 5.1.

More generally, our work is also related to research that applies machine learning or optimization to create new descriptors or improve existing ones. We describe some of these approaches in section 5.2.

5.1. Quantization-based descriptors

5.1.1. Local Binary Patterns

Ahonen and Pietikinen (2009) proposed to view the difference $c - n_i$ of each neighbor pixel n_i with the center pixel c as the response of a particular filter centered on c. Under this view, the LBP operator is a coarse way to quantize the joint responses of various filters (one for each neighbor n_i). Likewise, DTLBP and DLBP can also be seen as a quantizer of these joint responses, but it is built adaptively and discriminatively.

5.1.2. Trees and forests

Trees have become a popular quantization method in computer vision. Moosmann et al. (2008) use Extremely Randomized Clustering forests to create codebooks of SIFT descriptors (Lowe, 2004). Shotton et al. (2008) use random forests to create codebooks for use as features in image segmentation. While the use of trees in these works is similar to ours, they use the results of the quantization in a different way; the features are given to classifiers such as SVMs, which are not suitable for use in our problem.

Wright and Hua (2009) use unsupervised random forests to quantize SIFT-like descriptors for face recognition. The main difference with our algorithm is that we do not quantize complex descriptors extracted from the image. In addition, the accuracy of their algorithm

on the tested datasets is relatively poor compared to other state-of-the-art algorithms. This may be due to the use of an unsupervised algorithm to construct the trees.

5.1.3. Ferns

It is interesting to note that the simplification of trees to sets of comparisons is analogous to the simplification of random trees to random Ferns proposed for the task of keypoint matching (Ozuysal, Calonder, Lepetit, & Fua, 2010). Ozuysal et al. observe that Ferns give similar results to trees but have a smaller computational and space complexity. Though Ferns are similar in structure to our DLBPs, they are used differently - Ferns are used directly as multiclass classifiers (more specifically, to find the posterior probability of a class), whereas DLBPs are aggregated in histograms representing the properties of a local region.

5.1.4. K-means

There are various recent works using K-Means to construct codebooks to be used for face recognition in a framework similar to ours. Meng et al. (2006) use it to directly quantize patches from the grayscale image patches. Xie et al. (2009) as well as Lei et al. (2007) use it to quantize patches from images convolved with Gabor wavelets at various scales and orientations. These algorithms are close in spirit to our work, since they are partly inspired by LBPs. These algorithms differ from ours in the algorithm used to construct the codebook. They use K-Means, which has the drawback of not being supervised and thus unable to take advantage of labeled data. In addition, for the same number of codes, K-Means are less efficient than DLBPs and DTLBPs: in K-Means, quantizing a sample has linear complexity in the number of codes (K), whereas in DLBPs and DTLBPs the complexity is logarithmic, corresponding to the height of the tree in DTLBPs and to S in DLBPs. Finally, unlike ours, two of the above algorithms incorporate Gabor wavelet features; the cost of convolving the image with the real and imaginary parts of 40 or so Gabor filters may be excessive for some applications.

5.2. Other learning and optimization-based descriptors

5.2.1. Boosting

Our methods also differ from approaches that use techniques such as Boosting to select histograms corresponding to particular LBP windows or scales (G. Zhang, Huang, Li, Wang, & Wu, 2005; Liao et al., 2007) or histogram bins (Shan & Gritti, 2008; Wang, Zhang, & Zhang, 2009). The reason is that we do not select from among LBP features that have already been extracted, but instead search for the best feature to extract. Selecting from pre-extracted features is not feasible with a large number of pixel comparisons (S), since the length of the histograms grows exponentially with S.

5.2.2. Genetic programming

Another line of investigation worth mentioning is the use of heuristic algorithms, and in particular evolutionary algorithms, to construct visual descriptors for different purposes. Perez and Olague (2008) use Genetic Programming (GP) with a large set of terminals to construct invariant region descriptors for visual matching. Yu and Bhanu (2006) also use GP with a large set of operators and Gabor filtering to induce features for facial expression recognition. Kowaliw, Banzhaf, Kharma, and Harding (2009) use a variant of GP known as cellular GP to build features for an image classification task. Compared to these approaches, our features are simpler, since they do not use a complex set of operations and terminals.

5.2.3. Brown et al's discriminative descriptor learning

Of particular interest is the work by Brown, Hua, and Winder (2010), that systematically explores the design space of visual descriptors in the style of SIFT, SURF and HOG. They seek to maximize an empirical measure of descriptor discriminativity for keypoint matching. It would certainly be interesting to see this approach directly applied to face recognition. They descriptors from this approach are generally more computationally complex than our LBP-like descriptors.

6. CONCLUSIONS AND FUTURE WORK

We have proposed two methods that uses training data to create discriminative LBP-like descriptors. The first, based on decision trees, achieves superior accuracy compared to LBP and other LBP-like approaches, but at a large cost in size of the resulting descriptors. The second algorithm simplifies the tree structure and changes the learning algorithm to one better suited to create discriminative descriptors. This lead to descriptors that are more compact than DTLBPs and other LBP-like descriptors yet obtained superior or comparable results on standard face recognition datasets, specially when coupled with appropiate image normalization. This showed the advantages of using an adaptive and discriminative set of local binary patterns.

However, on datasets with large illumination variations our methods tends to underperform in comparison to methods that use Gabor filter banks. Incorporating Gabor filters (or other kind of filters) to DLBP is DTLBP is straightforward and remains a future avenue of research.

Another future line of research is the application of DLBP and DTLBP to other computer vision tasks, such as pedestrian detection, where local descriptors such as Histograms of Gradients have been succesful (Dalal & Triggs, 2005). This may require changes to the algorithm in order to accomodate the larger appearance variations that are seen in this datasets.

REFERENCES

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, *28*(12), 2037–2041.

Ahonen, T., & Pietikinen, M. (2009). Image description using joint distribution of filter bank responses. *Pattern Recognition Letters*, *30*(4), 368 - 376.

Albiol, A., Monzo, D., Martin, A., Sastre, J., & Albiol, A. (2008). Face recognition using HOG-EBGM. *Pattern Recognition Letters*, *29*(10), 1537–1543.

Bay, H., Ess, A., Tuytelaars, T., & Gool, L. V. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, *110*, 346–359.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.*, *19*(7), 711–720.

Bicego, M., Lagorio, A., Grosso, E., & Tistarelli, M. (2006). On the use of SIFT features for face authentication. In *CVPR* (p. 35).

Bolme, D., Beveridge, J., Teixeira, M., & Draper, B. (2003). The CSU face identification evaluation system: Its purpose, features and structure. In *ICCV*.

Bosch, A., Zisserman, A., & Muñoz, X. (2007). Image classification using random forests and ferns. In *ICCV*.

Brown, M., Hua, G., & Winder, S. (2010). Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*(99), 1. (Early Access)

Butman, M., & Goldberger, J. (2008). Face recognition using classification-based linear projections. *EURASIP Journal on Advances in Signal Processing*.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *CVPR*.

Dreuw, P., Steingrube, P., Hanselmann, H., & Ney, H. (2009). SURF-Face: face recognition under viewpoint consistency constraints. In *BMVC*.

Duda, R., Hart, P., & Stork, D. (2000). *Pattern classification*. Wiley.

Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., et al. (2008). The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Trans. Syst., Man, Cybern. A*, *38*(1), 149–161.

Gilad-Bachrach, R., Navot, A., & Tishby, N. (2004). Margin based feature selection - theory and algorithms. In *ICML*.

He, X., Yan, S., Hu, Y., Niyogi, P., & Zhang, H.-J. (2005). Face recognition using laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.*

Heikkilä, M., Pietikäinen, M., & Schmid, C. (2009). Description of interest regions with local binary patterns. *Pattern Recognition*, *42*(3), 425–436.

Kowaliw, T., Banzhaf, W., Kharma, N., & Harding, S. (2009). Evolving novel image features using genetic programming-based image transforms. In *CEC* (p. 2502-2507).

Lei, Z., Li, S., Chu, R., & Zhu, X. (2007). Face recognition with local Gabor textons. *Advances in Biometrics*, 49–57.

Li, S., & Jain, A. (2005). *Handbook of face recognition*. Springer.

Liao, S., & Chung, A. C. S. (2007). Face recognition by using elongated local binary patterns with average maximum distance gradient magnitude. In *ACCV* (pp. 672–679). Berlin, Heidelberg.

Liao, S., Zhu, X., Lei, Z., Zhang, L., & Li, S. (2007). Learning multi-scale block local binary patterns for face recognition. In *Advances in biometrics* (pp. 828–837).

Lowe, D. G. (2004). Distinctive image features from Scale-Invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Maturana, D., Mery, D., & Soto, A. (2010a). Face recognition with decision tree-based local binary patterns. In *ACCV*.

Maturana, D., Mery, D., & Soto, A. (2010b). Face recognition with optimized tree-structured local binary patterns. In *CWPR*.

Maturana, D., Mery, D., & Soto, A. (2011). Face recognition with discriminative local binary patterns. In *AFGR*.

Meng, X., Shan, S., Chen, X., & Gao, W. (2006). Local visual primitives (LVP) for face modelling and recognition. In *ICPR*.

Mikolajczyk, K., & Schmid, C. (2005). Performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(10), 1615–30.

Moosmann, F., Nowak, E., & Jurie, F. (2008). Randomized clustering forests for image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, *30*(9), 1632–1646.

Ojala, T., Pietikinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, *29*(1), 51–59.

Ozuysal, M., Calonder, M., Lepetit, V., & Fua, P. (2010). Fast keypoint recognition using random ferns. *IEEE Trans. Pattern Anal. Mach. Intell.*, *32*(3), 448–461.

Pentland, A., Moghaddam, B., & Starner. (1994). View-based and modular eigenspaces for face recognition. In *CVPR*.

Perez, C. B., & Olague, G. (2008). Learning invariant region descriptor operators with genetic programming and the f-measure. In *ICPR* (pp. 1–4).

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for Face-Recognition algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, *22*(10), 1090–1104.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, *1*(1), 81–106.

Ruiz-del-Solar, J., Verschae, R., & Correa, M. (2009). Recognition of faces in unconstrained environments: A comparative study. *EURASIP Journal on Advances in Signal Processing*, *2009*, 1–20.

Shan, C., & Gritti, T. (2008). Learning discriminative LBP-histogram bins for facial expression recognition. In *Bmvc*.

Shotton, J., Johnson, M., & Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *CVPR*.

Tan, X., & Triggs, B. (2010, June). Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE Transactions on Image Processing*, *19*(6), 1635–50.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71–86.

Wang, X., & Tang, X. (2006). Random sampling for subspace face recogntion. *International Journal of Computer Vision*, *70*.

Wang, X., Zhang, C., & Zhang, Z. (2009). Boosted multi-task learning for face verification with applications to web image and video search. In *CVPR*.

Wechsler, H. (2006). *Reliable face recognition methods: System design, implementation and evaluation*. Springer.

Wiskott, L., Fellous, J., Kruger, N., & Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, *19*, 775–779.

Wolf, L., Hassner, T., & Taigman, Y. (2008, October). Descriptor based methods in the wild. In *Real-life images workshop at ECCV*.

Wright, J., & Hua, G. (2009). Implicit elastic matching with random projections for pose-variant face recognition. In *CVPR* (pp. 1502–1509).

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.*, *31*.

Xie, S., Shan, S., Chen, X., & Gao, W. (2008). V-LGBP: Volume based Local Gabor Binary Patterns for face representation and recognition. In *ICPR*.

Xie, S., Shan, S., Chen, X., Meng, X., & Gao, W. (2009). Learned local gabor patterns for face representation and recognition. *Signal Processing*, *89*(12), 2333 - 2344.

Yu, J., & Bhanu, B. (2006). Evolutionary feature synthesis for facial expression recognition. *Pattern Recogn. Lett.*, *27*(11), 1289–1298.

Zhang, B., Shan, S., Chen, X., & Gao, W. (2007). Histogram of gabor phase patterns (HGPP): A novel object representation approach for face recognition. *IEEE Trans. Image Process.*, *16*(1), 57–68.

Zhang, G., Huang, X., Li, S., Wang, Y., & Wu, X. (2005). Boosting local binary pattern (lbp)-based face recognition. In *Advances in biometric person authentication* (Vol. 3338, p. 179-186). Springer Berlin / Heidelberg.

Zhang, W., Shan, S., Gao, W., Chen, X., & Zhang, H. (2005). Local gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition. In *ICCV*.

Zou, J., Ji, Q., & Nagy, G. (2007). A comparative study of local matching approach for face recognition. *IEEE Trans. Image Process.*, *16*(10), 2617–2628.