IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 23, 2017, accepted November 28, 2017, date of publication December 18, 2017, date of current version March 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2784842

INVITED PAPER

Face Feature Extraction: A Complete Review

HONGJUN WANG, JIANI HU, AND WEIHONG DENG
Beijing University of Posts and Telecommunications, Beijing 100088, China
Corresponding author: Weihong Deng (whdeng@bupt.edu.cn)

ABSTRACT Feature extraction is vital for face recognition. In this paper, we focus on the general feature extraction framework for robust face recognition. We collect about 300 papers regarding face feature extraction. While some works apply handcrafted features, other works employ statistical learning methods. We believe that a general framework for face feature extraction consists of four major components: filtering, encoding, spatial pooling, and holistic representation. We analyze each component in detail. Each component could be applied in a task with multiple levels. Then, we provide a brief review of deep learning networks, which can be seen as a hierarchical extension of the framework above. Finally, we provide a detailed performance comparison of various features on LFW and FERET face database.

INDEX TERMS Face recognition, feature extraction, filtering, feature encoding, feature aggregation.

I. INTRODUCTION

The ultimate aim of precise and automatic face recognition is essential for certain fields such as forensic science, automatic payment system, etc. In [1], face recognition is applied to unconstrained scenarios. The task is rather challenging and is still an open problem due to high variability such as in illumination, scales, pose, and occlusion. To achieve the object of robust face recognition, various approaches have been made by extracting overcomplete and high-dimensional local features from images and integrate them via learning algorithms to handle large data variations and noises. Bag-of-Features (BoF) model extracts local descriptors and then encodes them with a codebook (or dictionary) generated by machine learning techniques. Local features are encoded and classification is performed [2].

In this paper, our pipeline for facial feature representation is shown in Figure I.The pipeline is based on BoF model, but alterations have been made to cater for facial images.

This pipeline consists of 4 major components: filtering, encoding, spatial pooling and holistic representation. We analyze these features as follows:

- Filtering: to generate robust features for a given face image, it is beneficial to convolve the image with a specific filter, either using a pre-defined pattern, or using a discriminative filter learned from training dataset. The filtering could be multi-level to form cascaded filter image features. Also, as filtering of image patches can result in a huge collection of features which are prone to noise and variations, sometimes we apply quantization to compress local features to save computation time and storage. Many classic handcrafted local features combined filtering with quantization and histogramming techniques to form a robust mid-level feature.
- Encoding: The output of the encoding phase could be a histogram or a feature vector. Many primitive works encode local patterns via histogramming and thus do not explicitly contain a dictionary, while many others do. Dictionary and assignments can be generated via soft or hard clustering algorithms. There could be multiple levels of encoder.
- Spatial pooling: Spatial pooling can be seen as a way to further compress the coding vector according to the

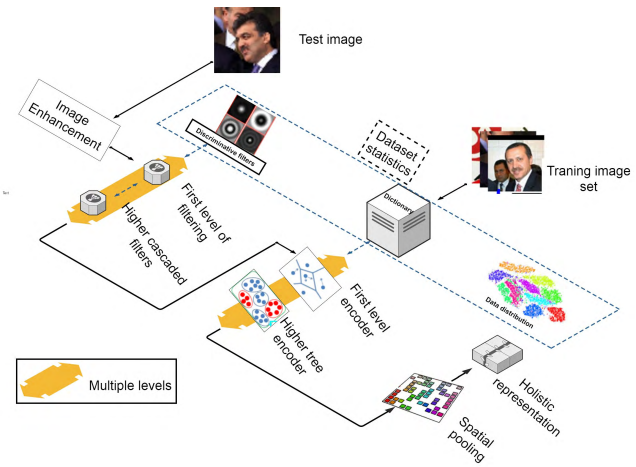


FIGURE 1. General framework for face feature extraction.

spatial layout of the image to form a final holistic feature. There are two classical pooling methods: average pooling that preserves the average response, and max pooling that preserve the maximum response. Block division can be seen as a part of feature pooling, as features extracted from different blocks are aggregated. This could be quite beneficial to face recognition, as the frontal structure of a face includes several distinct areas (eyes, nose, mouth, etc.). Facial appearance is largely based on those parts and is vital for recognition. Without proper division, local features extracted from different parts could be mingled together as we learn discriminant features based on those ones.

- Holistic representation: Many classical face recognition papers are mainly based on the holistic representation of the face. An extremely simple method is the template matching method for face recognition [3] that involved the direct comparison of pixel intensity values taken from facial images. Later, *Eigenface* method [4] calculated the eigenvectors and eigenvalues of the covariance matrix of facial images and only the principal components were preserved and compared. The *Fisherface* method [5] used both principal component analysis and linear discriminant analysis to produce a subspace projection matrix. The Fisherface method took advantage of within-class information by minimizing variation within each class, yet still maximizing class separation. However, for many recent face features, holistic representation is about dimensional reduction and feature fusion of the final features. Concatenation of block-based features could cause extremely high-dimensional features and therefore proper dimensional reduction is needed. Besides, to combine multiple features together, we have to resort to decision-level feature fusion techniques.

In Section II we discuss various filtering techniques, including Gabor, Difference of Gaussians, etc. We also analyze a specific form of local feature quantization and encoding based on histogramming. In Section III, we classify feature encoders with a codebook provided into several categories. They are: encoders based on K-means, Gaussian mixture models, sparse coding and tree models. Section IV is on spatial pooling, where block division, multiscale feature extraction and feature transforms are discussed. We discuss features regarding the whole face image in Section V, where we provide a deep analysis on works regarding feature transform, selection, and fusion. Note that these categories are not mutually exclusive, proper combinations of categories often yield decent performances. Towards the end of the paper, Section VII analyzes the popular deep learning frameworks. We provide intuitions and the reasons behind their stunning performances and show their frameworks and alterations made for face recognition. Section VIII compares performances and intricacies of various feature extraction approaches on FERET and LFW databases.

There are some surveys regarding face recognition conducted on variations of local descriptors like Gabor [6] and LBP [6]–[10]. Ahonen and Pietikäinen [11] presented an evaluation of different filters and quantization based on LBP. However, in those surveys, feature encoding and aggregation procedures were elementary and not robust. We would see that without proper algorithms, features and noises would be mingled together in a high-dimensional array, making it difficult to classify them properly. Recently, Huang *et al.* [2] did a theoretical survey on feature encoders and analyzed their motivations from a different point of view. A recent survey on pose-invariant face recognition gives several approaches to classify faces under extreme poses [12]. Their works enlighten us to do a review on face feature extraction.

II. FILTERING

In this section, we study the image texture and some commonly filtering techniques to describe face features locally. First, in Section II-A we discuss Gabor filters. In Section II-B we first discuss the widely used LBP descriptor. LBP is itself a filter (to extract pixel difference), a quantizer (binarize the difference and sum them up) and an encoder (generate histograms). In Section II-C we discuss histogram generation based on local textures. In Section II-C non-histogramming features like SIFT and HOG. Section II-D provides some inner relationships and potential combinations of discussed features. Section II-E discusses some of the preprocessing procedures. Section II-F lists handcrafted and adaptive filtering techniques. Lastly in Section II-G we list practices that realize the combination of various features by cascading.

A. GABOR FILTERS

Gabor wavelets are widely used in image processing field in that they capture local structure corresponding to spatial frequency, spatial localization as well as orientation selectivity. The definition of Gabor kernel is: $\phi_{\mu,\nu} = \frac{k_{\mu,\nu}^2}{\sigma^2}\exp(\frac{k_{\mu,\nu}^2 z^2}{2\sigma^2})[\exp(ik_{\mu,\nu}z) - \exp(-\frac{\sigma^2}{2})]$, where$\mu$ and ν are orientation and scale of the Gabor kernels respectively, and $z = (x, y)$ signifies spatial location. Wave vector is defined as $k_{\mu,\nu} = k_\nu e^{i\phi_m u}$ where $k_\nu = k_{\max}/f^\mu$ and $\phi_\mu = \pi\mu/8$ with $k_{\max}$ being the maximum frequency and f being the spacing factor between kernels in the frequency domain. Often, $\nu \in \{0, 1, 2, 3, 4\}$ and $\mu \in \{0, 1, 2, 3, 4, 5, 6, 7\}$, resulting 40 Gabor wavelets at 5 scales and 8 orientations, with $\sigma = 2\pi$, $k_{\max} = \pi/2$ and $f = \sqrt{2}$. A Gabor representation is obtained by convolving the input image with a set of Gabor filters of various scales and orientations, and it is also known to favor identity-related cues [13]. Moreover, the representation is robust to registration errors to an extent as the filters are smooth and the magnitude of filtered images is robust to small translation and rotations. However, convolution with a large number of filters makes Gabor filtering computationally costly and high dimensionality of the convolution output renders a dimensionality reduction step essential.

As Gabor phase information is sensitive to misalignment, many papers only extracted Gabor amplitude. Some practices

are: concatenating magnitudes of all orientations and scales into an augmented Gabor feature image and apply discriminative dimensional reduction with Enhanced Fisher Linear Discriminant Classifier(to be discussed in Section V-A) [14]; selecting most discriminative features via random forest [15]; constructing a Gabor image patch set for each Gabor kernel and then applied clustering approaches to each patch set to constitute a codebook to encode patterns via histograms [16]; extracting Gabor magnitudes on certain points determined based on a 3D deformable model [17]; using magnitudes for sparse representation and coding [18]; using magnitudes and then maximized squares of intra-face correlations via Partial Least Squares [19].

Though phase features are often ignored, we observe that with proper quantization techniques, phase information is beneficial for several face recognition scenario. A study by Wang *et al.* [20] quantized phase into bins according to the number of phase ranges and generate histograms based on co-occurrence of phase patterns. Another instance features Zhang *et al.* [21] that put forward Histogram of Gabor Phase Patterns (HGPP) to integrate Gabor phase information into the encoding scheme. HGPP was formulated with global Gabor phases and local Gabor phases encoded with phase-quadrant demodulation coding.

There are other ways of utilizing the Gabor filters. Full (real and imaginary) Gabor response are used in [22] and [23]. In [22], imaginary and real parts of Gabor response are used respectively to derive Local Gabor Textons, then they were labeled accordingly and a histogram sequence is calculated. Meyers and Wolf [23], on the contrary, do max pooling of various neighborhood scales to get a down-sampled representation. Cascaded feature (often combined with LBP) would be discussed in Section II-G. *V1-like* [24] models are composed of normalized, thresholded and clipped Gabor wavelet responses. The models are named after V1 cortex, or primary visual cortex. There were total 96 Gabors chosen to span an exhaustive cross of 16 orientations and 6 spatial frequencies. Based on this model, authors elaborated several other features by resizing, enlarging spatial frequencies and orientations covered.

B. LOCAL BINARY PATTERN (LBP)

The computation procedure of LBP is simple yet efficient: pixels of an image is labeled by thresholding the neighborhood of the pixel *locally* compared to the pixel itself to a *binary* number. Specifically, its feature vector is built by comparing the pixel with each of its neighboring pixels, and it interpolates values bilinearly at non-integer coordinates. We could use notation (P, R) to signify LBP parameters, which stands for extracting P sampling points on a circle of radius of R. The LBP operator uses a $(2R+1) \times (2R+1)$ kernel to summarize the local image structure. At a given center, (x_c, y_c), it takes the $(2R+1) \times (2R+1)$ neighboring pixels surrounding of the center pixel. However, R is often assigned to 1, which results in 8 neighboring pixels excluding the center itself. If the center pixel's value is greater than the neighbor's value, mark it with "1"; mark it with "0" otherwise. An 8-digit binary number suffice depicting local texture pattern. The pattern is then transformed into a decimal number by multiplying each digit by powers of two and sum. Given an image I and denoting i_c as the grey level of the pixel c of the image I, the LBP operator on this pixel is defined as: $LBP_{(P,R)} = \sum_{p=0}^{P-1} s(i_p - i_c)2^p$, with $s(x) = \begin{cases} 1 & \text{if } x \geq 0; \\ 0 & \text{otherwise.} \end{cases}$ For each image, the histogram is computed showing the frequency of each occurring number. For LBP codes with 8 neighbors, the histogram should have $2^8 = 256$ bins, thus the range of LBP is from 0 to 255. An important property of the LBP operator is its *invariance to monotonic photometric change* caused probably by illumination variations.

Though LBP could be directly used in face recognition, there are approaches to form a robust feature based on LBP. Feature fusion is one perticular example: images blocks are often used and histograms over cells are fused. However, another extension worth noting is the *uniform pattern* [25]: $LBP_{P,R}^{uniform} = \begin{cases} \sum_{p=0}^{P-1} s(i_p - i_c) & \text{if } U(LBP_{P,R}) \leq 2; \\ P+1 & \text{otherwise.} \end{cases}$, where U is a uniform measure defined as the number of spatial transitions (or bitwise 0/1 or 1/0 changes) in the binary pattern. According to this formula, LBPs with U value up to 2 are defined as *uniform patterns* and their labels are accumulated respectively (there are 58 of them), while non-uniform patterns are grouped into a single bin in the histogram. This idea originated from the fact that certain binary patterns occur more commonly in texture images than others. LBPs assign separate bins for every uniform pattern in a histogram, while all non-uniform patterns are assigned to a single bin. With uniform patterns, the length of the histogram with 8 neighbors reduce from 256 to 59, and the code is robust to noise. Uniform LBP is widely used in face recognition [26]–[33].

C. LOCAL FEATURES BASED ON HISTOGRAM

The pipeline of many local features are extremely similar to LBPs: apply a filter to the image; use quantization techniques and then obtain a histogram to represent the final feature. We now analyze these features in detail.

A most straight-forward filter could be the difference between pixel values of a center and its neighbor, or Pixel difference vectors (PDV). It is applied in [34]–[37]. Lu *et al.* [34], [35] calculated pixel differences for 48 neighbors within the 7×7 patch surrounding each pixel in the image, and learn an adaptive projection matrix to project the difference, and finally binarization via thresholding is applied. Lei *et al.* [36] obtained a collection of $s \times s$ pixel difference matrices and stretch them into PDVs. PDVs extracted would undergo a reduction in both dimensions and the number of features. Finally K-means clustering and labeling is performed for reduced features.

A drawback of LBP is its liability to local intensity variations such as noise and small wear-able ornaments. Many local features based on histogram attempt to alleviate the

impact of noises. We list them below:

- Refined features: Extended LBP [38] modeled absolute values of differences of the central point and its neighboring pixels. The difference vector of the 8-neighbor local descriptor $d = [d_0, d_1, ..., d_7]^T$ was normalized into [0, 1] as $\tilde{d}_i = (d_i - d_{min})/(d_{max} - d_{min})$.Values larger than 0.5 was set to 1 and otherwise set to 0. In this way, a supplemental code to the original LBP was obtained as a compact 8-bit binary string. Completed LBP (CLBP) [39] represented a local region by its center pixel as well as the sign and magnitude of local intensity difference. To effectively represent the magnitude information in LBP style, the center pixel was simply thresholded by the average gray level of the whole image as $s(x) = \begin{cases} 1 & \text{if } x \geq c_l; \\ 0 & \text{otherwise.} \end{cases}$, where c_l is the average gray level of the whole image. Similarly, the difference magnitudes were determined adaptively via a non-linear function $s(x) = \begin{cases} 1 & \text{if } x \geq m_p; \\ 0 & \text{otherwise.} \end{cases}$, where m_p is the mean value of the difference x. Local Directional Pattern [40] representation captured relative edge response value in eight directions at each pixel in the image. Local Gradient Pattern (LGP) [41] generated constant patterns irrespective of local intensity variations along edges and had a greater ability than LBP to determine the difference between face histograms. At a given center pixel position, gradient values between the center pixel and its neighboring pixels were defined as $g_n = |x_n - x_c|$, and the average of gradient values was set as $\bar{g} = \frac{\sum_{p=0}^{P-1} g_p}{P}$. LGP was defined as $LGP_{(P,R)} = \sum_{p=0}^{P-1} s(g_p - \bar{g})2^p$. Each bit of LGP was assigned '1' if the neighboring *gradient* of a given pixel was greater than the average of eight neighboring gradients, and '0' otherwise. LGP was later used in [27] as a local feature extraction step before feeding it to feature selection procedure.
- Refined sampling methods: In order to reduce the impact of noise, some papers compared not image intensity but the average intensity of image *blocks* [26], [42], [43]; or used multiple radius and patterns to sample neighboring pixels [36], [44]–[46]; or used statistical tools to adaptive determine sampling radius and pattern [47]. Local Quantized Patterns (LQP) [48] used not only a larger local neighborhood but also deeper quantization.
- Refined quantization: Local Ternary Patterns (LTP) [49] applied a 3-valued coding that had two thresholds around zero for improved resistance to noise. Resulting code can be treated as two separate channels of LBP descriptors, and LTP inherited high computational efficiency of LBP. Later an adaptive threshold for LTP using mean and standard deviation of the local region was put forward [50]. Some researchers based their encoder on a N-nary coding quantization scheme instead of binary coding method to preserve more structure information [51]. Other than LTP, Noise-Resistant LBP [52] introduced an *uncertain state* for small pixel differences, and attempted to determine the state with uniform patterns. We observe that several papers applied *soft histogram boundaries* with fuzzy membership functions [53], and later an adaptive version was put forward to determine the margin of decision boundary with histogram statistics [54].
- Refined procedure: Dominant LBP sorted occurring frequency of all generated LBP of the training data in descending order and found a number of patterns accounting for 80% pattern occurrences; in order to interact with distant pixels, they based their features on Gabor filter responses. Another example is the patch-level dual-cross patterns (DCP) [55] as a computationally efficient variation of LBP that is extremely robust to pose and expression variations. DCPs comprehensively yet efficiently encoded the invariant characteristics of a face image from multiple levels into patterns, and, via second-order statistic extraction and grouping, it is highly discriminative of inter-personal differences but robust to intra-personal variations. It was used in [56] for pose-invariant face recognition.

We now discuss other features. Local Phase Quantization (LPQ) was proposed by Ojansivu and Heikkilä [57] for blur insensitive texture classification through local Fourier transformation. However, it gained popularity due to its performance to non-blurred images and is often considered as a replacement of LBP in face recognition. The performance improvement may partially due to the size of local description, as LBPs are usually extracted from smaller regions with 3-pixel diameter, whereas LPQs are extracted from larger regions [58]. LPQ uses local phase information extracted using Short Term Fourier Transform (STFT) computed over a local $M \times M$ rectangle neighborhoods $\mathcal{N}_{\mathbf{x}}$. At each pixel location x, STFT of the image $f(x)$ is defined as $F(\mathbf{u}, \mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{N}_{\mathbf{x}}} f(\mathbf{x} - \mathbf{y}) e^{-j2\pi \mathbf{u}^T \mathbf{y}} = \mathbf{w}_{\mathbf{u}}^T \mathbf{f}_{\mathbf{x}}$, where $\mathbf{w}_{\mathbf{u}}$ is the basis vector of STFT at frequency u, and $\mathbf{f}_{\mathbf{x}}$ is another vector containing all $M \times M$ pixels from $\mathcal{N}_{\mathbf{x}}$. The local Fourier coefficients are computed at four frequency points: $\mathbf{u}_1 = [a, 0]^T$,$\mathbf{u}_2 = [a, a]^T$,$\mathbf{u}_1 = [0, a]^T$,$\mathbf{u}_1 = [a, -a]^T$, where a is a frequency parameter (a sufficient small scalar). For each pixel position this results in a vector $F_{\mathbf{x}}^c = [F(\mathbf{u_1}, \mathbf{x}), F(\mathbf{u_2}, \mathbf{x}), F(\mathbf{u_3}, \mathbf{x}), F(\mathbf{u_4}, \mathbf{x})]$. The phase information in the Fourier coefficients is recorded by examining the signs of the real and imaginary parts of each component in $F_{\mathbf{x}}^c$. In detail, let $F_{\mathbf{x}} = [\Re\{F_{\mathbf{x}}^c\}, \Im\{F_{\mathbf{x}}^c\}]^T$, and for each component f_j in $F_{\mathbf{x}}$, $LPQ_(x) = \sum_{p=1}^{8} f_j 2^{p-1}$. Similarly to an LBP, an LPQ describes the local neighborhood using an integer ranging from 0 to 255, and simply by counting LPQ patterns result in the local histograms with dimensionality 256. LPQ is useful in face recognition. Chan *et al.* [59] extended their Multi-scale LBP histogram in [60] to multi-scale LPQ histograms. Another experimental study on CMU-PIE, YALE-B and CAS-PEAL-R1 databases showed that low-level

LPQ descriptor is effective for blurred as well as sharp images [61].

Modified census transform (MCT) [62] operates on intensity mean of neighborhood and takes all 9 pixels in the 3×3 neighborhood (including the center). Denote intensity mean as $\bar{i}_c = \frac{\sum_{p=0}^{8} i_n}{9}$, MCT value for the center is calculated as $MCT = \sum_{p=0}^{8} s(i_p - \bar{i})2^p$. Chakraborti and Chatterjee [27] used LBP, MCT and LGP as a local feature extraction procedure. Jeong *et al.* [26] extended the sampling radius and could be applied to alleviate sensitivity of both LBP and MCT to noise.

Scale-invariant feature transform (SIFT) [63] has the merit of invariance to translations, rotations and scaling transformations in the image domain and robustness to moderate perspective transformations and illumination variations. The original SIFT descriptor firstly detect *interest points* from a gray-level image and then statistics of local gradient directions of image intensities were accumulated to give a summarizing description of the local image structures in a local neighborhood around each interest point. SIFT descriptor is computed as follows: it chooses a 16×16-pixel grid of local image patch around a point and then divides it into 4×4 windows. Image gradients are computed over the 16×16 array of locations in the image domain, and then the gradient directions are quantized into 8 discrete directions. During the accumulation of the histograms, the increments in the histogram bins are weighted by the gradient magnitude. Further, to give stronger weights to gradient orientations near the interest point, the entries in the histogram are also weighed by a Gaussian window function centered at the interest point and with its size proportional to the detection scale of the interest point. Thus SIFT descriptor is a local histogram computed at the 4×4 windows with 8 quantized directions, which lead to an image descriptor with $4 \times 4 \times 8 = 128$ dimensions for each interest point. To obtain contrast invariance, the SIFT descriptor is normalized to unit sum. In this way, the weighted entries in the histogram will be invariant under local affine transformations of the image intensities around the interest point, which improves the robustness of the image descriptor under illumination variations. Later, initiated in a scene classification application [64], the Dense SIFT descriptor that applies SIFT at dense grids which have been shown to lead to better performance. A basic explanation for this is that a larger set of local image descriptors computed over a dense grid usually provide more information than corresponding descriptors evaluated at a much sparser set of image points. Hu *et al.* [65] extracted several forms of local features: uniform LBP of 8×15 non-overlapping blocks, Dense SIFT descriptors on each 16×16 patch without overlapping to obtain 45 SIFT descriptors; Sparse SIFT computed at the nine fixed landmarks with three different scales. Local descriptors of the same kind were concatenated to form a global feature vector, forming three feature vectors for each image. They were used to train a deep neural network for metric learning, and score fusion was performed. However, unlike [65] that concatenated local features directly, Dense SIFT descriptors are usually accompanied with a clustering stage, where the individual SIFT descriptors are reduced to a smaller vocabulary of visual words and can then be combined with a bag-of-words (BoW) model or related methods. Originally K-means clustering (refer to Section III-B) are popular in generating codebooks, yet recent papers tend to use soft assignment clustering like sparse coding and Gaussian Mixture Model (GMM). We briefly review some methods and will discuss in detail in Section III-A. For example, Yang *et al.* [66] extracted Dense SIFT, and computed a spatial-pyramid image representation based on sparse codes of SIFT features, instead of the K-means and vector quantization. Motivated by the fact that kernel trick can capture the nonlinear similarity of features, Gao *et al.* [67] provided an extension by incorporating Kernel Sparse Representation with Spatial Pyramid Matching. Initiated in [68], Fisher vector described features via a normalized residual with respect to Gaussian cluster centers of GMM, this was later used in many papers [69]–[73]. Baecchi *et al.* [74] clustered densely extracted SIFT with Random Density Forest (RDF), an unsupervised method to minimize the Gaussian differential entropy of each split appears in as an alternative to GMM.

Speeded Up Robust Features (SURF) [75] is a variation of SIFT and it is several times faster than SIFT and claimed by its authors to be more robust against different image transformations than SIFT. To detect interest points, SURF uses an integer approximation of the determinant of Hessian blob detector, which can be computed using a precomputed integral image. The interest area is divided into 4×4 subareas that are described by the values of a wavelet response in x and y directions. To describe the subarea, the components involved in the calculations are $\{\sum dx, \sum |dx|, \sum dy, \sum |dy|\}$. SURF is used in [69] as a feature extractor.

Inspired by the SIFT descriptor, Dalal and Triggs [76] proposed Histograms of oriented gradients (HOG), a similar descriptor based on gradient orientation histograms computed over a grid of scale space. However, unlike the *local* image descriptor SIFT, HOG is computed within a region in a sliding window fashion and has the benefit of semi-global representation. The basic HOG includes the following computation: divide the image into smaller cells and compute edges of the image with the Canny edge detector; compute the orientation of each edge pixel; generate histogram by summing up the gradients having orientations in a certain range over every pixel within each cell in an image. In contrast to the SIFT descriptor, the HOG descriptor is not normalized with respect to orientation and thus does not possess rotational invariance. However, the histograms in the HOG operator are normalized with respect to image contrast. A variation of the method includes 2D HOG [77]. The algorithm employed the HOG which was reformulated in a 2D representation. Each bin represented one of the desired angles and was represented in the matrix, where the spatial relations were maintained and dealt with separately. HOG is widely used

in facial expression recognition. Albiol *et al.* [78] localized a set of 25 facial landmarks using the Elastic Bunch Graph Matching framework. The HOG features extracted from the vicinity in each of these 25 facial landmarks were used for classification, using nearest neighbor and Euclidean distance. Arguing that performance of [78] may crucially depend on the reliability of the landmark localizations, Déniz *et al.* [10] proposed to extract HOG descriptors from a regular grid in order to compensate for errors in facial feature detection due to occlusions, pose and illumination changes. Similarly, Berg and Belhumeur [79] extracted HOG, color histogram as well as an 8-bin gradient direction histogram from each grid cell and concatenated histograms over all cells. Another similar approach was to apply multi-kernel learning using HOG and LBP descriptor extracted in a regular grid manner [80].

2D Discrete Fourier Transform (DFT) could be used in face representation to provides a tool to remove unnecessary parts of frequency features [81]. Analyzing the face model in Fourier frequency domain provides more chances to pick out useful features on the assumption that we have known whether frequency bands are important or not. In the paper, Fourier features were learned independently from diverse frequency bands. Authors extracted Fourier features from three different Fourier domains: real and imaginary component domain, Fourier magnitude domain, and phase angle domain. Fourier spectrum is easily used to as a compensator for the phase shift faced with small spatial displacements caused by misalignment, and on the other hand, the complex phase spectrum is invariant to illumination variations and is tolerant to occlusion problems. This was later integrated into the proposed band selection scheme by choosing lower frequency information of a face model and then higher frequency information for analyzing more detail contours. The proposed band selections basically included the lower frequency, but on the other hand, they had different higher frequency bandwidths because higher bandwidths in company with the lower frequency have more discriminative information for detail facial components. Discrete Cosine Transform (DCT) decomposition is similar to DFT and it acts like a low-pass filter [82], [83]. Another similar Local Walsh-Hadamard Transform [84] applies a small size of Walsh-Hadamard Transform (WHT) to each pixel of an image by sliding the transform on the image pixel by pixel. The transform is believed to be robust to local variations and is later integrated to form a histogram by calculating phase-magnitude relationship.

We briefly introduce other methods:

- Higher-order derivatives: a high-order local pattern descriptor could encode local high-order derivative variations. While LBP can be viewed as a nondirectional first-order local pattern, second-order LDP can capture the change of derivative directions among local neighbors by encoding the *turning point* in a given direction [85]–[87].
- Code co-occurrence: LBP co-occurrence [88], HOG co-occurrence [89], or co-occurrence matrices based on quantized Gaussian phase information, and the frequency of each co-occurrence pattern with a fixed distance and along a specified direction were used to obtain the feature histogram [20].
- Similarity between oriented gradients: capture local geometric structure between a pixel and its neighborhoods by calculating the similarity between oriented gradients without quantization [90]. The descriptor is rather redundant so dimension reduction is utilized, then logistic function is used to binarize the feature.
- Feature interdependence relationship: considering each local region inside a face image as a graph vertex, set up an undirected connected graph capturing feature interdependence based information shared among the vertices [91].
- Binary optimization: Compact Binary Face Descriptor [34] is an optimization scheme for feature mapping to obtain *binary* discriminative features based on PDV. Optimization objectives were: maximizing the variance of binary code to obtain a compact code; minimizing binary quantization loss; ensuring even distribution of feature bins in the learned binary codes. To make more data-adaptive representation, the code was subsequently clustered and pooled. In their recent paper [35], clustering was integrated into binary discriminative mapping to form Simultaneous Local Binary Feature Learning and Encoding (SLBFLE) that could learn projection matrix, dictionary and coefficient matrix in an iterative manner and finally represented histogram feature with a learned dictionary.
- Feature selection: binarized pixel difference features are strongly correlated, thus feature selection could be made based on conditional mutual information [92]

We state again that the methods above are not mutually exclusive and proper combination could yield decent performance. Murala's paper [86] can serve as an illustration: it used average image intensity within a block to combat noise; the encoder was a three-bit derivative pattern along four defined directions, resulting in a 12-bit representation. For robust representation, pattern related to the magnitude of the center pixel was calculated and the two histograms were combined to obtain final feature. The framework of [87] is quite similar.

D. COMPLEMENTARY FEATURE EXTRACTION

We have introduced many ways to extract local features. Readers may easily find that these features, rather than mutually exclusive, are often utilized simultaneously to accomplish tasks. Features may be extracted and integrated in parallel, or in a cascaded manner.

We would like to mention backgrounds of individual features. Gabor filters, which are spatially localized and selective to spatial orientations and scales, are comparable to the receptive fields of simple cells in the mammalian visual cortex. Because of their biological relevance and computational

properties, Gabor filters have been adopted in face recognition. Since Gabor filters detect amplitude-invariant spatial frequencies of pixel gray values, they are known to be robust to illumination changes. LBP is widely used in analyzing textures, as it labels each pixel through binary gray scale difference of its 8 neighbors. The stunning performance of the descriptor in texture classification leads to its popularity and many variations and improvements are put forward. LBP is, by definition, invariant under any monotonic transformation of the pixel gray values. However, LBP is significantly affected by non-monotonic gray value transformations. Unfortunately, in contrast to the flat surfaces where texture images are usually captured, faces are not flat, therefore non-monotonic gray value transformations (manifested by shadows and bright spots) occur, and their positions could be changed depending on the illumination. LBP can be expected to have problems dealing with illumination variations in face recognition. Unlike LBP, SIFT and HOG depict the point in feature space by its weighted spatial histogram of gradients of its neighbors. Both SIFT and HOG are orientation-based, robust to brightness changes, and computation initiatives of the two descriptors for individual pixels are similar. SIFT is often combined with a detector of interest point to match local regions of interest. However, it has been shown that densely extracted SIFT descriptors are helpful in recognition and thus it is extracted in a Bag-of-words fashion and widely used in computer vision.

Some papers compare the performance of these descriptors on their feature extraction algorithms. Lu *et al.* [28] verified the performance of their proposed Random Path measure (to be discussed in Section IV-A) with four popular descriptors in face recognition: LBP, HOG, Gabor, and a statistically learned descriptor. Chen *et al.* [29] investigated the effect of the dimensionality of the feature on face verification accuracy. They tested on LBP, SIFT, HOG, Gabor to find out high-dimensional feature resulted in high performance. There was a $6\% - 7\%$ improvement in accuracy when increasing the dimensionality from 1K to over 100K for all descriptors. Recently, Zhu *et al.* [30] employed the over-complete high-dimensional features proposed in [29], including high-dimensional Gabor and LBP as face representation as well as a pose and expression normalization method for face recognition.

Local features like LBP, SIFT or HOG only capture texture or gradients in a small region and may ignore holistic characteristics. In addition, one single feature suffers from the insufficiency in describing discriminant structures for classification. Thus these features may be considered as complementary and in Section II-G and Section IV-C we see many encoding practices that integrate one feature (for instance LBP) with other features in a cascaded way or fusion on a certain level and apply voting for the final decision. In practice, a sensible combination of complementary local patterns could be beneficial for generating robust features to tackle face recognition problems, especially on datasets with large variation in illumination, background, pose, etc.

E. PREPROCESSING

With a fixed-sized face image, it is a common practice to start extracting local features right away. However, our facial images at hand may contain a large portion of background irrelevant to our objective. Thus in practice, our first step is to crop the face part out. Face detectors such as the Voila-Jones detector [93] is mostly used to locate landmarks and accordingly crop out (and rescale to) a uniformly sized face part. In addition, approaches in handling lighting variations have been studied and readers may refer to [94] for a comprehensive understanding in illumination preprocessing for face recognition. Some preprocessing procedures are observed in several papers.

- Downsampling: downsample the image before LBP extraction to enhance execution time [95].
- Filtering: preprocessed images with Difference-of-Gaussians (DoG) filter for their proposed LBP-like descriptor [44], [45]. Applying DoG could remove high-frequency noise and low-frequency illumination variations, thus providing a robust result.
- Color space transform: Liu and Liu [96] based their descriptor on a novel hybrid RC_rQ color space that is constructed out of RGB, YC_bC_r and YIQ color spaces. Component images in the novel color space possess complementary characteristics and enhance the discriminating power for face recognition.

F. OTHER FILTER DESIGN TECHNIQUES

In this section we briefly discuss filters that use other encoding methods other than histogramming. Relative intensity contrast between neighbors utilized by LBP could be an inadequate filter in some scenarios, so many papers try to grasp more discriminative features than intensity contrast.

- prominent directions: edge information could be beneficial to classification, and edge response prominences of different directions could be used. Local Directional Patterns (LDP) [97] computed the edge response values in all eight directions at each pixel position and generated a code from the relative strength magnitude. Given a central pixel in the image, applying the Kirsch compass edge detector could obtain eight edge response values $m_0, m_1, ..., m_7$. By finding the top k values of $|m_i|$ and set them to 1, and by setting the remaining $8 - k$ bits of 8-bit LDP pattern to 0, the LDP code was calculated as follows: $LDP_k = \sum_{p=0}^{7} s(m_p - m_k)2^p$. Zhong and Zhang [98] refined the procedure above by taking into account the distinction between most and second-most prominent edge response directions.
- rectangle filters: Jones and Viola [99] used a set of computationally efficient rectangle filters to describe local features. Each features measured input images at particular locations, scales and orientations. Later Adaboost was used to select discriminative features.

There are works on obtaining a discriminative image filter via statistical learning. A simple supervised machine learning

utility, Fisher Separation Criteria (FSC), argues that discriminability is achieved by maximizing the ratio of *between-class scatter* and *within-class scatter*. FSC is also the criterion function to be maximized in Linear Discriminant Analysis (LDA), a supervised dimension reduction method. FSC is widely used for learning discriminative mapping as well as discriminative sampling procedures. For example, FSC could help choose from many neighborhoods the most discriminative ones the central pixel to compare with [36], [38], and [100]; to help determine the value of respective filters via optimization [36], [38]; approximately solve non-linear kernel functional mapping of image patches [101].

Beside FSC-based, other discriminative sampling and filtering discriminative mapping and sampling observed are:

- Binarized Statistical Image Features (BSIF) [102] could describe a pixel's neighborhood by a binary code obtained by convolving the image with a set of linear filters and then binarization. Unlike LBP and LPQ, the set of filters in BSIF was learned from a training set by maximizing the statistical independence of the filter responses. Each bit of BSIF was associated with certain filter and the value of each bit in BSIF code string was computed by binarizing the response of a linear filter with a threshold at zero. To be precise, it projected image patches to a subspace which basis vectors were learned from Independent Component Analysis, then binarizes them by thresholding. The bits in the code string corresponded to binarized responses of different filters.
- Observing that severe self-occlusion hampers block-based methods, Ding *et al.* [56] proposed Patch-based Partial Representation by extract facial textures from unoccluded blocks only. Each block adopted Multi-task Learning to learn transformation dictionary that transformed the features of different poses into a common discriminative subspace to enhance recognition ability. Separate transformation dictionary was learned for each patch, thus the number of resulting transformation dictionaries equated to the number of blocks.

G. CASCADED FEATURES

LBP is a texture encoder that encodes intensity difference between pixels. However, our images at hand may subject to noise or large variations, so it is beneficial to use its histogram on other features. On the other hand, Gabor feature could capture the local structure corresponding to specific frequency, spatial locality and selective orientation and it has been demonstrated to be robust to noise, illumination and expression changes. Therefore many researchers attempted to apply LBP on Gabor-like features rather than the pixel intensity to obtain sufficient and stable representations.

Among papers, the most popular could be LBP on multi-orientation and multi-scale Gabor magnitudes, which is named Local Gabor Binary Pattern (LGBP) [103]. Its merit over LBP lies in its capacity of representing face images as many spatial histograms with varying orientations and scales. It is believed that Gabor feature and LBP characterize the property of local texture distributions in distinct and complementary ways, thus combining them can be beneficial. The feature histogram is obtained by the following steps: convolve the input face image with 40 multi-orientation and multi-scale Gabor filters to obtain 40 Gabor Magnitude Pictures (GMPs) in frequency domain; each GMP was further encoded by LBP and converted to binary LGBP Maps; each LGBP Map was further divided into non-overlapping rectangle regions with specific size, and histogram was computed for each region; LGBP histograms of all the LGBP Maps were concatenated to form the final histogram sequence as the model of the face. Later authors proposed to encode real and imaginary Gabor Feature Map with LBP histograms based on exactly the same procedure [104] Similar studies include: performing LBP on multi-orientation log-Gabor image in order to be robust to illumination and expression changes [105]; performing LPQ and LBP fusion on Gabor features, so that the feature could utilize blur invariant property and texture information to recognizing blurred or low-resolution face images [106]; performing LQP on Gabor filtered images and utilizing *vector quantization* and *lookup table* to refine quantization while keeping low computational complexity [107].

Lei *et al.* [108] made a generalization of previous approaches by viewing Gabor faces as a stack where three axes X, Y, T respectively denotes *rows*, *columns* of face images and *different types of Gabor filters*. Existing methods applied LBP on XY plane, while authors proposed Gabor Volume LBP (GVLBP) by conducting analysis on XT and YT planes: projecting neighboring relationship of different face images filtered by Gabor of various orientations and scales. The projected feature lay in three 2D spaces: one image space (XY), and two spaces that account for variations of different face images (XT and YT) and image space coordinates XY respectively. The paper also proposed an effective version (E-GV-LBP) to encode features from three domains *simultaneously* (the 8 neighborhoods consists of: 2 orientation neighboring pixels, 2 scale neighboring ones and 4 neighboring pixels in spatial domains). In light of uniform LBP, authors proposed statistical uniform pattern to generate histograms. Though E-GV-LBP representation was more efficient than GV-LBP-TOP, it was still of a huge dimension and its redundancy greatly affected the efficiency in feature matching process. Thus authors later utilized Conditional Mutual Information for feature selection and then LDA for feature transformation.

Inspired by HGPP [21] (as we mentioned in Section II-A), two attempts were made [109], [110] to integrate Gabor phases with Gabor magnitudes. They argued that though Gabor phases are sensitive to local variations, they could discriminate between patterns with similar magnitudes and provide more detailed information about the local image features. The two works differentiate mainly after calculating the posterior to the encoding scheme: Zhang *et al.* [109] matched features with histogram

intersection, while Guo *et al.* [110] resorted to sparse coding. Experiments also demonstrated that encoded phases, unlike their magnitude counterpart, are resistant to illumination [109]. Xie *et al.* [111] encoded Gabor phase by XOR operator. To alleviate the sensitivity of Gabor phase to the varying positions, phases were firstly quantized into a different range, two phases were believed similar local features if they belonged to the same interval and reflected different local features otherwise.

Apart from Gabor, there are other possible alternatives to feed into an LBP encoder.

- Using color information to combat illumination variations: independently extracting LBP from each color channel [112]; introducing color vectors at each pixel location within each of the local face regions of multiple spectral-band images [113].
- Using derivatives of Gaussian-like function $G(x, y) = e^{\frac{-(x^2+y^2)}{\sigma^2}}$ to avoid redundancy of Gabor feature: feeding normalized first and second order derivatives to LBP [114]; exploiting the first derivative of Gaussian to form a multi-directional pattern for DCP [55]. Experiments demonstrated that first and second order derivatives were excellent descriptors for features such as bars, blobs and corners in images and higher order features could describe more complicated structures but were difficult to exploit because of their sensitivity to noise.
- Decomposition of face image using curvelet transform: applying logarithm transform and LBP encoder to the lowest frequency band to represent face structure, and normalizing other frequency bands to reflect edge structure changes [115]. Both logarithm transform and normalization were done in order to remove dominance of specific values.
- Sobel filter: enhancing edge information by building LBP on Sobel filtered faces [116]. Gabor real/imaginary parts could also be done prior to Sobel operation to form G_Sobel-LBP.
- Adaptive filters: learning the filter via FSC in a supervised way to achieve discriminate ability and encoding with LBP [38].
- Similar to LGBP, Patterns of Oriented Edge Magnitudes (POEM) [117] encoded relationships between local edge *magnitude* distributions through different orientations. The algorithm was proved to have low computational complexity compared to LGBP and thus capable of high-performance real-time face recognition. Vu and Caplier [118], authors fused POEM with a complementary PDO (patterns of dominant orientations) descriptor. Unlike POEM that considered the relationship between edge *magnitude* distributions, PDO encoded the relationships between dominant *orientation* of local image patches by firstly calculated and assigned the block's dominant orientation to its central pixel, and then encoded the dominant orientation using an operator. In a later paper [119], the authors put forward a complementary Patterns of Orientation Difference (POD) descriptor to capture the relationship between *orientation* of image patches. POD differed from PDO in the sense that it did not require estimating the dominant orientation of patch and thus it both avoided possible estimation errors and reduced the computational complexity.

TABLE 1. Cascaded features.

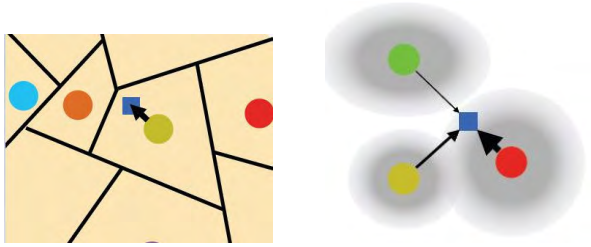
Encoding Phase 1	Encoding Phase 2	Reference
Gabor magnitude	LBP	[121], [122], [123], [124], [104], [103]
Gabor magnitude + phase	LBP	[109], [110]
Log-Gabor	LBP	[105]
Gabor magnitude	LBP/LQP/LTP	[107]
Local edge through different orientations	LBP	[117]
Sobel or Gabor + Sobel	LBP	[116]
Gabor Volume	LBP	[108]
Discriminantly filtered image	LBP	[38]

Juefei-Xu and Savvides [120] studied the real-world scenarios where only a partial face is captured or instances when only the eye region of a face is visible, especially for the cases of uncooperative and non-cooperative subjects. They encoded coefficients of a variety of discrete transforms including Walsh-Hadamard Transform (DWT), Discrete Cosine Transform (DCT), Discrete Hartley Transform (DHT) and discrete polynomial transforms. LBP on these frequency domain representations had much richer and more discriminative information than spatial-domain representation. And later they employed several subspace representations (principal component analysis (PCA), unsupervised discriminant projection (UDP), kernel class-dependence feature analysis (KCFA), and kernel discriminant analysis (KDA)) on them to match the periocular region on a large data set such as NISTs Face Recognition Grand Challenge (FRGC) ver2.0 database. Verification results on periocular images matching showed the merit of discrete transforms over mere LBP.

We selectively summarize papers cascading various local features in Table 1.

III. FEATURE ENCODING

In the previous section, we have discussed the filters and the generation of histograms without a codebook. Histograms without a codebook are easily implemented yet they could not adapt to specific dataset. In this section, we would discuss a more general way of encoding filter responses. Our discussions focus on four facets: encoders based on K-means, encoders based on GMM, sparse representation and tree-based encoder. The encoders are often based on codebooks learned. Section III-B, Section III-C deals with encoders on K-means and GMM generated codebooks respectively. Section III-D is about Sparse representation, an algorithm that differentiates from the above methods in encoding and codebook generation. Lastly we discuss about a quantizer and encoder that utilize a *tree-like graph* in Section III-E.



(a) (b)

FIGURE 2. Example encoder based on codebook generated by K-means and GMM. (a) Hard partitioning. (b) Soft assignment.

A. ENCODERS BASED ON CODEBOOKS

Encoders based on codebooks can be split to a training phase, where we learn a codebook, and an encoding phase, where the dictionary is used to extract features from new inputs. Expectation Maximization (EM) algorithm is an iterative method for finding maximum likelihood and is commonly used for codebook generation. One can form learned feature vectors based on these codebooks. There are two forms model for doing this: the *discriminative* model, which is based on hard partitioning, and the *generative* one, which soft assign labels according to probability affinity. An extremely popular hard version of EM algorithm is *K-means* algorithm, which locates cluster centers by finding centers of Voronoi cells. This method of representing features via cluster centers resembles lossy compression methods and is often named *vector quantization* (VQ): a mapping from the original feature space to a finite set of codebooks (representative vectors). As for soft EM, Gaussian Mixture Model (GMM) is a codebook learning algorithm obtained by probability assignment. With generated codebooks, encoding procedures ensue. In practice, we could use K-means as an initialization, followed by EM algorithm for GMM training. Figure 2 is an illustration of an encoder for K-mean and GMM codebooks. Figure 2 is the core idea of Vector of Locally Aggregated Descriptors (VLAD), which is a learned feature computed by summing the residuals of local feature and each cluster centers found by K-means. Figure 2, in contrast, illustrates the formulation of Fisher Vector (FV), which consists of the summed first and second order statistics regarding local features and each GMM centers. FV formulation is more of a probabilistic approximation. Its intuitive is: the gradient of the log-likelihood describes the direction in which parameters should be modified to best fit the data [125]. We would discuss encoder based on K-means and GMM, and lastly Sparse Representation would be introduced for local features as a special form of encoder, where codebooks and their corresponding coefficients are found to reconstruct the image under sparsity constraints.

A generalization of vector quantization discussed above is the tree-based encoder. Two commonly used tree-based encoding frameworks are: *decision tree* and *projection tree*. A decision tree can be viewed as a tree-like graph or model of decisions and their possible consequences as a predictive model. It can form *classification trees* by mapping observations about an item to conclusions about the item's target variable, which can take a finite set of values (their respective labels). Projection tree, on the other hand, hierarchically partition the source space into pieces in a manner that is provably sensitive to low-dimensional structure. It can be viewed as a novel method of VQ and variant of the k-d tree that is manifold-adaptive [126].

B. ENCODERS BASED ON K-MEANS

A widely used K-means form in texture analysis scenario is to convolve an image with N different filters, whose response at a certain pixel forms an N-dimensional vector. A collection of such vectors is clustered with certain algorithms to form a codebook. Then a histogram is built with each position of the image is labeled with nearest cluster center and the histogram of the image is used as a descriptor [11], [34], [36], [45], [107], [127]. These works clustered various local features: projected filter responses, LBP histograms, LTP histograms, etc. Of course, K-means could be used directly on image patches to obtain 'textons' for each face block, and features are labeled according to the textons to generate a histogram for face recognition [16], [22], [128]. There are some variations of this idea: using weighted histograms to signify the importance of face componenets [127]; dividing images into groups and K-means is performed twice: firstly within each group and then on its resulted centers [22].

K-means can be used to generate codebooks for conventional VLAD [129] model, which is done in Euclidean spaces with centers of Voronoi cells obtained via K-means. Assume we would like to encode an image with local features extracted as $X = \{x_t, t = 1, ..., T\}$, VLAD captures deviations of the distribution of local descriptors assigned to a cluster center, each local descriptor x_t is associated with its nearest visual word μ_k in the codebook, and all local descriptors associated with μ_k are aggregated as $v_k = \sum_{t=1}^{T} x_t - \mu_k$. VLAD has a final representation size of KD, with K the number of cluster centers and D feature dimension. There are practices that apply various local descriptors (SIFT and SURF) encoded by VLAD to face recognition field [69]. Contrary to conventional VLAD, Faraki *et al.* [130] extracted LBP locally and extended VLAD to an extensive space of curved Riemannian manifolds. Extension to Riemannian manifolds exploited geodesic distance to determine the closest descriptor to each codeword. Given the metric, the codebook was trained using an EM-based approach similar to standard K-means. Hard assignment K-means can also be used to facilitate computation efficiency of GMM models [70], [82].

C. ENCODERS BASED ON GMM

Contrast to hard assignment with K-means, generative clustering models with GMM are observed in some other researches. For example, Sharma *et al.* [37] modeled differential vectors with GMM instead of predefined hard quantization procedure, and image representation was based on Fisher score. The use of GMM to derive soft probabilistic

quantization on differential vectors could be seen as a generalization to LBP, as LBP can be viewed as a discretization of differential space into two bins per coordinate.

Probabilistic Elastic Part (PEP) is a representation by concatenating local descriptors of facial parts obtained through performing GMM on spatial-appearance features, the densely-sampled multi-scale local features augmented by the descriptors spatial location coordinate. Each Gaussian component stands for patches of a semantic meaning (e.g. eye, mouth). A testing image is represented in bag-of-words manner where the feature pair that induces the highest probability on each Gaussian component is found. In detail, to encode an image, local descriptors are extracted and each one of the K Gaussian components commits *one* descriptor with the highest probability, and PEP is formed by concatenation of such chosen descriptors, forming the representation of the face. In this method, GMM act as a bridge to establish the semantic correspondence between two images. One detail is that Gaussians in GMM is restricted to be spherical to enforce more localized Gaussian components. This method is present in three papers. Li *et al.* [71] used Universal Background Model and GMM to model image patches. Video face recognition system of Li *et al.* [72] represented features as the mean of PEP across all video frames to naturally suppress appearance variations. This representation is followed by PCA and joint Bayesian classifier. Li and Hua [73] applied PEP model hierarchically to present a deep model. Face image was hierarchically decomposed into face parts of different levels, exploiting fine-grained structures of face parts and capturing pose-invariant sub-structures. The model located face parts and sub-parts in a top-down manner and then construct representation with SIFT from bottom-up either supervisedly or unsupervisedly.

FV is also an important model for face recognition. The central idea of FV encoding is to aggregate higher order statistics of each codebook learned into a high dimensional vector. More specifically, a GMM is trained as the visual codebook. Then, averaged first-order and second-order distance statistics with respect to each Gaussian component is concatenated to form the final feature representation. For local descriptors of an image $X = \{x_t, t = 1, ..., T\}$ and GMM p, parameters of p is denoted as $\theta = \{w_k, \mu_k, \Sigma_k, i = 1, ..., K\}$, which denote prior, mean vector and covariance matrix of the k-th Gaussian, respectively. The gradient $\nabla_\theta \log p(X|\theta)$ is the average higher-order statistics between a D-dimensional image descriptor and a center of GMM. The likelihood that x_t is generated by the GMM is $p(x_t|\theta) = \sum_{k=1}^{K} w_k p_k(x_t|\theta)$, in which $p_k(x_t|\theta) = \frac{\exp\{-(x-\mu_k)'\Sigma_k^{-1}(x-\mu_k)/2\}}{(2\pi)^{D/2}|\Sigma_k|^{1/2}}$ and $\sum_{k=1}^{K} w_k = 1$. We denote the probability for feature x_t to have been generated by the k-th Gaussian as $\gamma_t(k) = \frac{w_k p_k(x_t|\theta)}{\sum_{j=1}^{N} \log w_j p_j(x_t|\theta)}$. We calculate the gradient of x_t with respect to GMM center and covariance as: $\mathcal{G}^X_{\mu_k} = \frac{1}{\sqrt{w_k}} \sum_t \gamma_t(k) \frac{x_t - \mu_k}{\sigma_k}$, $\mathcal{G}^X_{\sigma_k} = \frac{1}{\sqrt{2w_k}} \sum_t \gamma_t(k) [\frac{(x_t-\mu_k)^2}{\sigma_k^2} - 1]$, and they are 1st and 2nd order statistics, respectively. Concatenation of the two and we obtain $2DK$-dimensional FV. Simonyan *et al.* [68] first used FV to represent dense SIFT features augmented by normalized descriptor location. To construct a more compact and discriminative face representation, they proposed an efficient dimension reduction method for high-dimensional FV. Recently, Li *et al.* [131] proposed a selective encoding framework that injected foreground/background probability into each cluster of a descriptor codebook. An additional selector component also discarded distractive image patches and improved spatial robustness. The scheme was applied to FV features to alleviate inaccurate face localization in unconstrained face recognition problems. FV could also encode RDF [74]. RDF can be seen as a generalization of GMM that create different partitions of the feature space. The split used at each node of each tree is determined by a random sampling process. Given a set of input SIFT descriptors, authors randomly generated a fixed number of candidate splits and selected the best split parameter for each node by maximizing an information gain function. This paper originated from the major drawback of the enormous size of the final image descriptor, and it offered a tradeoff between dimensionality and precision.

Encoders based on GMM may be disadvantageous to cover complex datasets in that the number of Gaussians need to be specified manually. A discriminative feature extractor named GaussianFace [132] used multi-task learning based Discriminative Gaussian Process Latent Variable Model (DGPLVM) that based on the *non-parametric* Gaussian Process Latent Variable Model [133]. DGPLVM mapped a high dimensional data representation to lower-dimensional latent space using a *discriminative prior* on latent variables to encourage latent positions of the same class to be close and different class to be far while maximizing the likelihood of the latent variables in the Gaussian process framework. The resulting covariance matrix could be used to group latent data points automatically. After latent representation of the data, the following steps were similar to classical feature extractor: first-order and second-order statistics to the centers were computed; Gaussian process binary classifier was employed to obtain corresponding probability and variance; statistics and variance were concatenated and represented as high-dimensional facial features. Further, to alleviate the complexity of DGPLVM, they introduced approximation and a computationally more efficient equivalent form of Kernel Fisher Discriminant Analysis to DGPLVM to simplify calculations.

D. SPARSE CODING

Since each patch only captures subtle facial textons, many patches are visually similar to each other. Directly using the intensity feature for face recognition may not be favorable, because even a tiny perturbation like expressions or noises might induce a considerable change on the distances between two patches. To improve the robustness and enhance the discriminability, many papers employ sparse representation

to generate powerful representations from low dimensional error-prone patches to high dimensional sparse coefficient space.

The underlying idea of sparse representation is to represent a query patch as a *sparse* linear combination of a dictionary, assuming that each subject has sufficient samples in the dictionary to span over possible subspaces. Therefore, each probe image patch can be considered to be represented by a sparse code α that is comprised of coefficients that linearly reconstruct the patch via the dictionary A. Sparse representation can be seen as a special form of soft assignment with codebooks learned. Sparse coding relies on the sparsity assumption, and the assumption holds when each class in the gallery has sufficient samples and the query lies on the subspace spanned by the gallery of the same class. Only those atoms in the dictionary that truly match the class query sample contribute to the sparse code. In practice, sparse representation with ℓ_1-minimization approach that solves an optimization problem based on reconstruction error is often implemented. It reconstructs a given patch as a linear combination of dictionary atoms. This section majorly deals with sparse representation used as feature generation for each image blocks, and sparse representation used to generate a global feature would be discussed in Section V-D.

In 2009, Wright *et al.* [134] systematically formulate an approach for Sparse Representation-based Face Classification. Given a test sample from one of the classes in the training set, sparse representation was computed and assignment to a specific class was done according to non-zero entries in the sparse coefficients α. Considering the presence of noise and variations, they assign the image to which resulting least residual using coefficients associated with that class. To reject invalid test images, authors devised a validation test procedure by defining a *sparsity concentration index* that quantify concentration (or spread) of sparse coefficients for different classes. They tested some holistic feature extraction method (Eigenface, Laplacian face, downsampled image and random projection) and argued that it is beneficial to reduce data dimension before sparse representation. They also tackled occlusion by introducing a novel Occlusion Dictionary whose basis matrix could be an identity matrix, given occlusion was rather local. The algorithm would seek the sparse solution on both the original dictionary and the Occlusion Dictionary. They focused on the sparse representation of the whole image and partial face features (eyes, nose, mouth, and chin), but found in the experiment that reconstructing with sparse coding independently for each image blocks achieved a better result.

As sparse representation linearly reconstruct a probe image by all the training images under sparsity regularization, its performance relies heavily on *dictionaries*, or *codebooks*. While K-means clustering is one possible choice of codebook generator [135], there are models that more commonly used and we list them below:

- K-SVD: K-SVD is a generalization of K-means, and is observed as a mean to learn over-complete visual dictionaries for residual-based face recognition. K-SVD algorithm could be used for densely sampled image patches [136], DCT features [83], or patches extracted at particular landmarks [137].
- FSC: Yang *et al.* [138] presented a supervised dictionary learning method based on FSC. The model exploited discriminative information in not only the representation residual but also the representation coefficients to classify query image. Dictionary atoms had correspondences to class labels, and FSC was enforced to sparse representation coefficients, thus good reconstruction was enforced to same class training samples and poor reconstruction was enforced to other classes.
- Intra-class dictionaries: Introduced by Deng *et al.* [139] (to be discussed in Section V-D), intra-class dictionaries expanded the training set and gained robustness, especially in Single Sample Per Person (SSPP) case. Zhu *et al.* [140] observed that different importances of facial parts contributed to the problem. They adopted a local representation approach: each patch of the query sample was represented by the local gallery dictionary and an intra-class variation dictionary at the corresponding location, and they aimed to minimize total residual. Gao *et al.* [141] also adopted intra-class variance dictionary to represent variances in illumination, expression, occlusion. In addition to intra-class variance dictionaries, they formed an optimization procedure considering 3 facets: non-zero coefficients only appeared at the places which corresponding to the person these patches belong to (by enforcing group sparsity constraint); each patch coefficients associated with intra-class variance dictionary should be sparse; reconstruction error should be small.
- Random dictionaries: Shen and Shen [135] experimented dictionaries generated by K-means, K-SVD or randomly of different sizes and sources to find that the choice of dictionary learning methods might not be important and learning multiple dictionaries using different low-level image features often improved the final classification accuracy. The paper also showed that spatial pyramid with max-pooling was beneficial and soft threshold encoding could achieve results on par with sparse representation and with less computation time.

We analyze several sparse coding examples. Min and Dugelay [142] divided face into 3 levels ($2 \times 2, 4 \times 4$ and 8×8 blocks each) like a pyramid. In each division level, sparse codes of LBP histograms were elementary-wise summed together, rather than concatenated. Similarly, we could select meaningful features from high-dimensional overcomplete LBPs via sparse representation [143]. In another sparse system, only non-negative codes are retained within each block: the nonnegative constraint on the code allowed direct sum a set of codes without considering their negative values [144].

Correlation of sparse codes could also be a useful measure of image similarity. Guo *et al.* [145] proposed a face

verification framework by extracting several local features, using correlation and dissimilarity of the sparse codes of the image feature pair to form similarity score, and performed score level fusion. On the other hand, Liu *et al.* [146] partitioned each face into a set of overlapped blocks and classify each block, then aggregated the classification results by voting to make the final decision. To generate classification result for each block, blocks were further divided into overlapped patches. By assuming image patches lie in a linear subspace, the central patch of the test block could be approximately represented by a linear combination of the patches in the corresponding block from the same class. This core assumption reflected local structure relationship of overlapped patches and makes sparse coding feasible for SSPP problem. Finally, they integrated all the classification results by voting. Theodorakopoulos *et al.* [147] observed that: in the sparse representation of patches corresponding to the same facial location of different pictures taken from the same individual, the majority of the atoms used would be the same, despite illumination changes or small expression variations. Thus, the Hamming distance between the sparse coefficient vectors derived from sparse coding procedure of the two patches would be an efficient local dissimilarity measure between the two facial images. Therefore, they utilized block-based sparse coding and Hamming distance to express pairwise similarities between faces. The authors also elaborated a new criterion for the rejection of person not registered in the database by estimating the ratio between the overall maximum similarity value and the maximum value achieved ignoring the values corresponding to the person that is matched with.

Collaborative Representation was initiated by relaxing sparsity constraint in sparse coding from ℓ_1 norm to ℓ_2 norm and solve by least square [148], [149]. It represented query sample with non-sparse ℓ_2-regularization rather than ℓ_1-regularization. Liu *et al.* [146] extended their SSPP sparse correlation coding scheme to collaborative representation for face recognition. To improve the performance of collaborative representation for small sample size problems, Zhu *et al.* [150] put forward Patch-based Collaborative Representation, where the representation was conducted on patches of different scales and classification was done by combining recognition output of all the overlapped patches. Ensemble learning was utilized to fuse information at different scales optimally. In [151], a locality-constrained version was adopted, in which the objective function coded the training data and its nearest neighborhood to produce minimal reconstruction errors simultaneously.

As a specific example, Wong *et al.* [83] showed a typical sparse-based face feature extraction procedure. Facial images were divided into blocks and transformed via local transformation like DCT. For each block, the sparse vector was generated by the sparse encoder using a learned dictionary. While sparse coding was employed as the encoder, they tackled an ℓ_1 minimization problem with dictionaries obtained by K-SVD. Each region was then described by a sparse code by average pooling strategy. The framework is extremely similar to [136], where aligned and geometrically normalized image was firstly filtered with DoG to get rid of illumination variations and noise. Next, with a handcrafted sampler, an intensity vector was obtained at each pixel. Then the vector was sparsely encoded to a non-negative code vector with an overcomplete dictionary learned offline by K-SVD on face patches. Sparse code vectors within a pre-defined block were *summed* together to form a descriptor of that block, and descriptors were concatenated to form the face descriptor. Compared with histogram based methods, the summation of sparse coefficients was similar to soft clustering and was more robust to variation in image appearance. Finally authors applied PCA to reduce dimension.

TABLE 2. Various sparse coding approaches.

Feature extraction	Reconstruction	Reference
Entire images or image patches	sparse coding for each image division	[134]
Gabor feature	sparse coding	[18]
LBP	semi-supervised sparse coding with identity constraint	[152]
Image patches	Nonnegative sparse coding and pooling	[144]
Local patch vector of blocks	sparse coding and collaborative representation	[146]
Multi-scale blocks	Patch-based collaborative representation fused by ensemble learning	[150]

As a summary, Table 2 shows some of the typical spase coding approaches.

E. TREE-BASED ENCODER

Here we list decision trees and projection trees which perform as a quantizer and encoder that transform local descriptor input and generate quantized code output.

A primitive study of the decision tree (DT) in face verification appeared in a paper of Nowak and Jurie [153], who used extremely randomized decision trees according to theoretical studies of [154]. In the scenario, only same or different labels were known. The paper extracted SIFT feature and quantized them using randomized trees for similarity measurement. Several pairs of corresponding local patches were sampled from a pair of images, with each patch pair assigned to several clusters with an ensemble of extremely randomized decision trees. The cluster memberships were combined with pre-determined optimized weights to make a global decision about the pair of images. Randomized trees saved training and testing time and also gave good properties when dealing with high-dimensional features. A variant of the random decision tree, Random Density Forest, is an unsupervised method to minimize the Gaussian differential entropy of each split that appeared in [74] as an alternative to GMM to create different partitions of the feature space as well as to create discriminative visual vocabularies. In the paper, FVs were built over the RDF (refer to Section III-C). Another approach by Maturana *et al.* [155] learned discriminative features by a supervised DT, forming DT-LBP. Rather than comparing pixel with all its neighbors, DT-LBP chose the most informative pixel comparison based on entropy impurity. The obtained LBPs were adaptive and discriminative in that: they were constructed by decision tree and randomized tree construction algorithms that had been shown to be very effective

in computer vision applications; constructing different tree for each region enabled different discriminative patterns for different face image regions; much larger neighborhood could be used and the algorithm could decide relevant neighbors. Authors had found that for their problem single trees were more effective than random trees. A hierarchical DT was built in [156] using a bottom-up approach by recursively clustering and merging classes at each level. This DT was used for feature selection from a set of potentially discriminative HOG features: for each branch of the tree a list of HOG features was built using the *log-likelihood maps* to favor locations that are expected to be more discriminative. Recently, Gong *et al.* [157] utilized DT followed by a new probabilistic matching framework. The encoder was such trained with face images that the frequency of output codes distributes as evenly as possible and discriminative ability in terms of maximum entropy is maximized. This could be seen as a robust and discriminative generalization of LBP algorithm.

Projection trees partition a feature space into cells in a recursive manner, splitting the data along one projection direction at a time. A binary tree whose leaves are individual cells was thus built with the succession of splits. The code of each test sample is assigned according to the cell (the leaf node) it belongs to. Wright and Hua [158] used random projection tree to quantize discriminative local descriptors. The resulting trees were only weakly data-dependent and exhibit good generalization in practice even across very different datasets. Cao *et al.* [45] encoded image via K-means, PCA tree and Random Projection (RP) tree for comparison and RP tree outperformed other methods. Zhang *et al.* [44] first extracted features with their LBP-like descriptor, then they put forward coupled information-theoretic encoding, which adopted Mutual Information Maximization between photos and sketches and achieved coupled encoding by coupled information-theoretic projection tree.

IV. SPATIAL POOLING

Previous sections focus on filters, quantization, and encoding. In this section, we discuss measures to obtain a *global* representation with the tool of *spatial pooling*, where many papers observed the facial image structure (with face parts like eye, nose, and mouth) as a prior and made divisions accordingly. Various feature pooling methods can be regarded as an enhancement to feature encoding.

Spatial pyramid [66] (as mentioned in Section III-D) is a systematical framework that can be seen as a generalization of a basic form of spatial pooling–*block division*. This is equivalent to applying descriptors in each separate blocks independently and respectively. Then, aggregate or concatenate all the representation vectors for each division to form a global one. This simple strategy is fast as well as efficient, thus is applied widely to face recognition experiments, and it inspired countless visual models including *Multiple Spatial Pooling* that describe the spatial structure with multiple Gaussian distributions with respect to local features' locations in the image space and setting the centers of blocks as Gaussian pooling clusters [159]. As block divisions of the spatial pyramid are done hierarchically to capture features of various scales. Therefore, descriptors of multiple scales and multi-descriptor fusion could be beneficial as well. We would discuss block division in Section IV-A and multiscale features in SectionIV-B, respectively. We would discuss a direct way to fuse features via concatenation in Section IV-C.

A. BLOCK DIVISION

Face is rather structural, if histogram is computed within the whole image domain, spatial facts of each feature would be lost. Therefore provided that face alignment was done, it is beneficial to define facial image blocks for us to extract descriptors from within separately, hence two descriptor will match only if it is extracted from the same block location. Through this way, important geometric information of face could be encoded within our descriptor. A direct approach done in most papers extract features on square, manually selected, non-overlapping blocks of same size respectively, normalize and concatenate them together, as done by [40], [59], [60], [80], [108], [109], [116], [147], [155], and [160]–[165]. Some papers provide details of their blocks, [117], [166] divided the image into 8×8 blocks, [97], [36], [44], [167], [110], [168] partitioned the image into 10×10, 7×7, 7×5, 5×5, 4×2, 3×3 blocks respectively. Lei *et al.* [22] divided the image into 7×8 regions. Meng *et al.* [128] showed relation between block number and accuracy in a table. Some works improved the simple divisions above, for example: practicing several partition schemes and fused them in a metric learning framework [144]; manually weighting facial blocks according to their significances [21], [32], [34], [89]; appending spatial coordinates of patch center to SIFT features to form an augmented local descriptor [68], [131]. Apart from non-overlapping blocks, overlapping block partition was adopted in [140] and [151]. Chan *et al.* [169] composed block-wise histograms with or without overlapping to form final features, and suggested that non-overlapping blocks are suitable for face images. Yuan *et al.* [170] explored several partition schemes (10×10, 5×5, 4×4, 2×2). Zhong and Zhang [98] utilized a 10×10 scheme for PIE database, and 8×8 for YALE. Chowdhury *et al.* [171] utilized non-overlapping sub-images and the whole image as well. Finally, contrast to regular square division, several papers proposed non-square division based on fitting 3D face models [172]–[174].

Data-adaptive learning approaches are made to obtain optimal block division. FSC is a relatively popular way of optimizing block weight to achieve discriminative recognition. Jiang *et al.* [175] used FSC for block weight optimization to achieved discriminant classification. Two experiments on FERET database [20], [21] utilized non-overlapping 8×8 blocks and weighted them by FSC. Lei *et al.* [22] and Zhang *et al.* [21] weighed LGT histograms and HGPP respectively based on FSC.

Another approach includes: Quad-tree was used in [101] to select best (overlapping and non-overlapping) block division. Authors went down the quad-tree stopping at the level beyond which no improvement in recognition rate is observed. It is also worthwhile to note that in the testing phase, each patch from the test image casted a vote towards image classes, and the class with maximum votes won.

Landmarks can be detected to give us a prior for choosing proper regions to extract locations, and many works extracted blocks surrounding each landmark specified only [45], [79], [152], [176]; or based their features on Elastic Bunch Graph Matching of landmarks [78]. A recent architecture [177] combined multiple features on multiple landmark patches together and normalized it to form a final feature for classification.

Inspired by spatial pyramid, many works applied hierarchical block division. Sanderson and Lovell [82] put forward the hierarchical Multi-Region Histogram (MRH) scheme by divide face into 2×2 regions, where each region was further divided into small overlapping patches. For each patch, descriptive features were placed into a vector. This 'visual word'-inspired overlapping MRH was used as a representation of 2D DCT features. Wong *et al.* [83] employed sparse coding within the MRH framework to form a novel descriptor. Each region in MRH was described by average pooling sparse codes within the region. Uzun-Per and Gokmen [84] divided the image into non-overlapping blocks and overlapping subregions in each block. To overcome the problem of unintentionally dividing meaningful parts into different subregions, the authors made shifting at the starting point and make two partitions. In a similar manner, Liu *et al.* [146] took a hierarchical division by partitioning the image into overlapping blocks and blocks into overlapping patches. The partition was aggregated by plurality voting, and the scheme was capable of multi-scale operation, though the paper experimented only the single-scale version. Sandereson *et al.* [178] later devised a multi-layer feature extraction procedure, in which overlapping, weighted blocks, and descriptors were pooled. Features were extracted separately and they devised a *boosting*-based method to learn salient region along with optimal mixing weight. Pairwise distance of training images for each region was combined to determine the most useful ones.

Rather than concatenating features within blocks, we observed some works analyzed feature interdependencies among blocks. Yao *et al.* [91] projected face image blocks onto an undirected connected graph and interdependencies between local regions were encoded, which lead to a new facial feature descriptor called Spatial Feature Interdependence Matrices. Lu *et al.* [28] observed the distributions of face images for one person may have different densities, shapes and proposed to measure *similarity* among patches by constructing networks based on random path (RP) measure for face recognition. RP measure includes all paths of different lengths in the network and could capture robust discriminative information. They divided faces into multiple overlapping patches of the same size and modeled them by constructing two face patch networks: the in-face network and the out-face network. The in-face network was constructed for *one pair of faces*: at each patch location, two corresponding patch pairs and their eight neighboring patches were used to form a graph, and patch similarity according to RP was calculated and integrated. The out-face network was constructed rather globally over the training space: for each patch, a search was conducted over the *whole database* of face patches and they found similar patches in the same location neighbors of the patch to form the patch pair network. The similarity of the two networks could be optimally combined to form a similarity vector for SVM classification. Kumar *et al.* [179] extended the naive block-based weighted plurality voting classification approach. They divided the image into 8×8 pixels *blocks*, classified each block with a classifier, took higher order relationships among blocks into account using *kernels*, then aggregated blocks by weighted plurality voting.

Block-wise concatenation of descriptors would lead to extremely high dimensional vectors. To alleviate this effect, discriminative features are needed. Common practices are feature transform and feature selection. PCA, an unsupervised feature transform method, can remove noises by discarding eigenvectors corresponding to small eigenvalues and was adopted in [29] and [92]. However, PCA is easily affected by high-frequency visual words with large eigenvalues. Especially, for face images, smooth facial areas like cheek and forehead area may lead to recurrence of same visual words. These high-frequent visual words contribute strong responses for the corresponding eigenvectors and eigenvalues when using PCA. A better way is to use Whitened PCA (WPCA) which suppresses the responses from larger eigenvalues. WPCA is observed in [144] and [180]. Xie *et al.* [111] introduced a block-based Fisher's linear discriminant (FLD) on LGBP and LGXP to reduce dimension while enhancing their discriminative powers. The basic idea of block-based FLD is to first blockwise divide local descriptor into multiple feature segments, apply LDA to each segment and combine the decisions of all block-wise features. BFLD was also utilized in [84] and [181]. Shen and Shen [135] applied ZCA and local contrast normalization to each patch. PCA could also be viewed as a filtering process. Chan *et al.* [169] utilized a cascade of PCA filter banks to form 'PCANet' features. It is followed by simple binarization, indexing, and pooling with respect to block-wise histograms to form the final sparse feature. Inspired by PCANet, Lei *et al.* [182] generalized their DFD descriptor [36] to a stacked image descriptor (SID). SID was optimized in a forward layer-wise way, thus could be seen as the pre-training result for Convolutional Neural Network (CNN). Four implementations including PCA-SID, Discriminant Tensor Analysis-SID (DTA-SID) and DFD-SID were introduced. Contrary to PCANet which views the response of different PCA channels as different samples, responses of deeper levels of PCA-SID were combined by different projective weights that were learned.

Moreover, integrating higher-order information for discriminant learning was possible via DTD-SID, and discriminant convolutional filter with optimal sampling was possible via DFD-SID.

Besides feature transform, there are discriminative feature selectors. Jones and Viola [99] selected important features using Adaboost. Adaboost is based on the following fact: using a single feature to classify can result in slightly better than random performance, so it can be used as a weak classifier. Adaboost learns the classification by selecting only those individual features that can best discriminate among classes. Zhai *et al.* [183] selected discriminating patch subset by *jointly* considering points, descriptor and similarity in a patch-descriptor-similarity space. Zini *et al.* [143] selected *meaningful* and *sparse* features simultaneously via Group LASSO. Group LASSO directly modeled multi-class problems and allows feature group selection (select the whole group or discard all features belonging to it) simultaneously discriminating among all the identities. Taking the computational expense of straightforward Group LASSO into consideration, Huang *et al.* [180] proposed a two-step metric learning method to enforce sparsity, to avoid features with little discriminability and to improve computational efficiency. The paper firstly selected groups of features with a Mahalanobis matrix, and then another Mahalanobis matrix was learned to exploit the correlations between the selected feature groups in the lower-dimensional subspace.

B. MULTI-SCALE LOCAL FEATURE GENERATION

Face feature matchings are often performed at different scale level. To accommodate objects of various scales, multi-scale block division is observed in many papers. Reference [14], [21], [22], [103], [104], [123], [124], and [164] extracted multi-scale Gabor features for non-overlapping blocks and concatenate them. Particularly, the number of scale of Gabor features in [14], [21], [22], [123], [164], and [124] were 5 and 3 respectively. Yang and Zhang [18] extracted local Gabor features of 5 scales and 8 orientations and performed sparse coding on these features. Some partitions involve weighting different blocks: [16], [103], [123] extracted multiple scales of Gabor features and calculates block weight according to FSC; Jeong and Kim [122] proposed a four-step weighting scheme of 6×5 blocks based on their inner variations. Meyers and Wolf [23] operated on image filtered by Gabor of various sizes, performed max-pooling, concatenated features of different scales and orientations and weighted them.

Spatial pyramid, a coarse-to-fine spatial division scheme, is also a popular choice for multiscale features [166], [184]. Galoogahi and Sim [185] applied spatial pyramid of 5 levels. Shen *et al.* [186] densely extracted multi-scale patches via spatial pyramid, built a multi-level pyramid and adopted multi-level spatial pooling. The spatial pyramid could also be combined with sparse coding frameworks [66], [67], [135].

LBP features can be also extracted in multiple scales by varying its parameter R [60], [88], [187]–[191], and multi-scale LBP (M-LBP) performed better than single scale LBP. Experiments in [192] had shown M-LBP on non-overlapped block division performed better than its overlapped version. Liao *et al.* [193] made an alternation of M-LBP, where the average sum of multi-scale blocks was thresholded. Lu and Tang [132] extracted M-LBP on overlapped patches surrounding *accurate landmarks only*. Gong *et al.* [157] divided images into overlapping patches and put forward a generalization of LBP matching based on maximum entropy and identity factor analysis. Still, recognition accuracy was improved with M-LBP on various scales integrated. A variant of M-LBP is Hierarchical M-LBP (HM-LBP) which extracted information from non-uniform LBP patterns without extra training [95], [194], [195], and the former two works extracted HM-LBP on 3 scales.

Apart from Gabor and LBP features which are direct ways to attain multi-resolution, other features appeared as well. For example: Ouamane *et al.* [32] attained LPQ of different scales with various window and filter sizes; Berg and Belhumeur [79] used two scales of grid for HOG feature; Ding *et al.* [56] extracted 3 scales of DCP feature. Lei *et al.* [196] had done experiments comparing Local Phase Quantization (LPQ) based on a 9-scale Short-term Fourier Transform with non-overlapped blocks and overlapped blocks and found the latter option better. Multi-scale patch extraction surrounding landmarks was seen in [29] and [137], where the former used 5-scale patch extraction on 27 accurate landmarks, and the latter 2-scale on 26 landmarks. Zhu *et al.* [150] classified query samples with collaborative representation by combining recognition outputs of 7-scale overlapping blocks. Similarly, Nan *et al.* [197] utilized collaborative representation for non-overlapping multi-resolution blocks and aggregated them by weight fusion.

Other innovations are mainly inspired by image pyramid. In [158], multi-scale dense patches all across the image were extracted from a Gaussian pyramid of 3 sizes. Jun and Kim [41] set up an image pyramid for face detection. Multi-scale block was also achieved by Bing *et al.* [198] with block division and image pyramid, whose weights were learned via Fisher criterion. Li *et al.* [71]–[73] combined image pyramid, overlapping patches and location augmentation to form spatial-appearance features. Struc *et al.* [199] had face image cropped tightly, normally and broadly according to landmarks to achieve a 3-scale presentation. In a similar fashion, Hwang *et al.* [81] formed 3 face models by cropping face to 3 levels: fine, middle and coarse. Huang *et al.* [180] enumerated rectangular regions of varying sizes from 8×8 to 96×144 within the 110×150 region. In a similar manner, Min and Dugelay [142] applied multiple block-division schemes, which resulted in blocks of multiple scales, and extracted LBP histograms for sparse representation and classification.

We selectively list methods that integrate spatial information and their respective characteristics in Table 3.

TABLE 3. Various block division approach.

Method	Weighted	Overlapping	Multi-scale	Landmarks oriented	Reference
Manually selected squares	No	No	No	No	[160], [117], [108]
Manually selected squares	Supervisedly learned	No	No	No	[200]
Quad-tree auto block selection	No	Both	No	No	[101]
Blockwise interdependency analysis	No	No	No	No	[91]
Landmark-based patch extraction	No	No	LBP-like radius	Landmark only	[45]
Spatial pyramid + similarity weighting	Pairwise FSC weighting	No	Spatial pyramid	No	[185], [43]
Histogram of concatenated M-LBP	No	No	Yes	No	[60]
LGBP	No	No	Gabor	No	[124]
LGBP + similarity weighting	Pairwise FSC weighting	No	Gabor	No	[123]
HM-LBP	No	No	HM-LBP	No	[194]
Block-wise histogram	No	Both	No	No	[169]
Manually selected squares	Manually	No	No	No	[89]
Manually selected squares	FSC	No	No	No	[20], [21]
Manually selected squares	No	Yes	No	No	[82]
Multi-layer (combines by pooling)	Boosting	Yes	No	No	[178]
Gabor ordinal measures	No	Both	Gabor	No	[164]
Patch-based Partial Representation	No	Yes	DCP	No	[56]
HOG pyramid	FSC	No	Image pyramid	No	[198]
HOG on regular grid	No	Yes	Multiscale HOG	No	[10]
Dense multiscale patches	No	No	Image pyramid	No	[158]
Spatial pyramid	No	Yes	Spatial pyramid	No	[186]
HOG	No	No	No	EBGM based on 25 landmarks	[78]
Image pyramid+location augmentation	No	Yes	Image pyramid	No	[71], [72], [73]

C. FEATURE-LEVEL FUSION

Fusion of different descriptors is a way to gain robust descriptors and it appeared in many works. The fusion of different modalities is generally performed at two levels: *feature level* and *decision level*. In the feature level fusion, the features extracted are first concatenated into a single feature vector and then sent to a classifier. Its advantage mainly lies in the simplicity of training (as only one learning phase on the combined feature vector is required) and the exploitability of correlation between multiple features at an early stage. However, it is required that features to be fused are represented in the same format before fusion. Some practices are: [201] combined variants of LBP to form a joint histogram; in [178], local LBP features and holistic Laplacianface features were extracted; in [202] several local descriptors including pixel coordinates, intensity and derivatives were used. Other examples are: LBP+LPQ [170], LBP+Gabor [203],LBP+LTP [167], LBP+pixels [135], LBP+grayscale and gradient images [46], LBP+SIFT [71], POEM+POD [119]. Some tricks are essential to guarantee a rational feature combination. Please note that PCA and normalization was performed in [203].

Hwang *et al.* [81] made use of three types of Fourier features extracted: concatenated real and imaginary components, Fourier spectrums and phase angles. To obtain more salient features, they adopted three different frequency bands designed for each individual feature, as well as three face models based on image scale. These features were concatenated, and subspace methods were used to alleviate increased dimensionality.

Our aforementioned practices are typical examples of feature-level fusion, a simple practice by concatenating features. Decision level and other types of fusion would be mentioned in Section V-C.

V. HOLISTIC ENCODING

This section discusses *holistic encoding*, which deals with the final feature vector for the whole image that learned from local features discussed in previous sections. In Section IV, descriptors obtained from various localities are aggregated to form global features.

Initiated in [64] and [204], computing SIFT over dense grids in the whole image domain, rather than finding several points of interest, is often preferred in experimental evaluations. Our observation shows that features in many experiments were multiple densely extract features concatenated together. Extracting patterns from huge amounts of multi-dimensional data can be overwhelming, and matching concatenated descriptors is time-consuming. Though some papers utilized block-wise dimension reduction or apply models like VLAD and FV to aggregate descriptors, the resulting descriptor can be still huge. Thus various kinds of transformations are often applied to reduce dimension, which can be observed in many papers. And here in Section V-A and V-B we discuss about some commonly used procedures to deal with the final feature. In Section V-C the discussion of fusing multiple features continues. Differentiated from Section IV-C that deals with feature level fusion, in this section we discuss decision level fusion and its variants. Lastly, Section V-D aggregates studies about the holistic encoding that still prevails nowadays: Global Sparse Representation.

A. FEATURE TRANSFORMATIONS

Subspace approaches are frequently applied to features or histograms to remove unreliable dimensions and to derive a compact representation. PCA is commonly found in papers to reduce feature (histograms) dimension by the nature of data distribution, as can be seen in [10], [17], [45], [80], [90], [162], [199], and [205]. Some

variations of PCA are observed: The employment of 2D PCA [206] maintained the spatial relationship between the pixels in the training images while allowed compact representation of the images which yielded excellent recognition speed and storage requirements. Inspired by 2D PCA, Abedelwahab *et al.* [77] introduced a 2D HOG representation and used 2D PCA to reduce the dimension of their 2D HOG representation. Ren *et al.* [207] applied Asymmetric PCA (APCA) to remove unreliable dimensions in feature space, as the number of inter-class sample pairs was much larger than the number of intra-class pairs in that problem. Some papers [65], [84], [208] utilized WPCA to histograms to increase its discriminative power and robustness.

LDA is a supervised feature transformation procedure that maximizes the ratio of *between class scatter* and *within class scatter*. Many works applied LDA to transform features into subspaces [10], [59], [60], [164], [171], [196]. With high dimensional weighted histogram features, Lei *et al.* [108] utilized Conditional Mutual Information (CMI) for feature selection and then LDA for feature transformation. CMI selected effective and uncorrelated feature set and LDA was adopted to learn discriminative feature space to improve efficiency and effectiveness.

The combination of PCA and LDA (PCA+LDA) is a simple yet powerful combination and is widely used in face recognition field. The method was proposed in [209] and it consists of two steps: first facial descriptors are projected to a subspace by PCA, second LDA is used to obtain a linear classifier. The rationale could be: given fewer data points than the dimension of data, within class scatter turns out to be a singular matrix and LDA could not be performed directly; further, matrix inversion is sensitive in high-dimensions, thus LDA tends to overfit the data. PCA constructs a task-specific subspace so that generalization ability of LDA is improved when only a few samples in each class are available for training. This combination was used in [44], [81], [123], and [163] for feature compression. To achieve age-invariant face recognition, Gong *et al.* [157] applied PCA+LDA to feature vector before feeding it to Identity Factor Analysis (IFA) for classification. IFA first decomposed features with respect to mean, identity-related component, age-related component, and noise and then estimated the probability that the two faces had the same underlying identity.

Variants of PCA and LDA is observed in a number of papers to cater for specific tasks.

- Tan and Triggs [203] offered a kernelized variant of LDA called Kernel Discriminative Common Vector (KDCV) to seek optimal discriminant subspace of fused features. KDCV used a nonlinear kernel mapping to implicitly transform input data to high dimensional feature space, then it selected and projected out an optimal set of discriminant vectors in the space, using the kernel trick to express resulting computation in terms of kernel values in the input space.
- Arashloo and Kittler [33] proposed a nonlinear binary Class-Specific Kernel Discriminant Analysis Classifier (CS-KDA) based on spectral regression kernel discriminant analysis. By using spectral regression, eigen-analysis computation in PCA and LDA was avoided. With CS-KDA, a regional discriminative face image representation was established with multi-scale local features.
- Enhanced Fisher Linear Discriminant Model (EFM) was introduced in [210] to improves the generalization capability of LDA by decomposing the LDA procedure into a simultaneous diagonalization of the two within- and between-class scatter matrices. EFM was preceded by PCA in [161] to reduce feature dimensions. Liu and Liu [96] used EFM for dimensional reduction of MLBP and as a classifier to obtain similarity score as well. Liu and Wechsler [14] applied EFM to augmented Gabor feature to derive low-dimensional features with enhanced discriminant power.
- Chakraborti and Chatterjee [27] extended Gravitational Search Algorithm (GSA), an LDA-like metaheuristic optimization algorithm, to a binarized version to compress LBP and LGP binary codes. The paper introduced a novel dynamic adaptation of weight features for GSA to tackle binary decision making problem in feature selection.
- Kumar *et al.* [101] recovered a Volterra *kernel* by minimizing the ratio between intra-class distances and inter-class distances, which was essentially the same to LDA.
- Several subspace representations (PCA, Unsupervised Discriminant Projection, Kernel Class-dependence Feature Analysis, Kernel Discriminant Analysis) was employed on Discrete Transform encoded LBP features for periocular matching application in [120].

Other feature transformation methods are:

- To tackle poor results combining Relevant Component Analysis (RCA) [211] with dimensionality reduction, Meyers and Wolf [23] invented a kernelized and regularized version of RCA to weigh local features as well as compress the high-dimensional concatenated feature, then they did normalization and square rooting.
- Zhou *et al.* [115] used locality preservation projection (LPP), which was modeled by a nearest neighbor graph that preserved image space local structure. LPP resulted in a face subspace for each individual and each face image was mapped into a low-dimensional face subspace, which was characterized by a set of feature images named Laplacianfaces.
- Simonyan *et al.* [68] proposed a linear feature transform based on discriminative metric learning. The low-rank linear projection of descriptors minimized the distance between images with of the same face and maximized it otherwise. Based on the above projection, Parkhi *et al.* [70] further decreased the number of bits required to encode face tracks with binary compression.
- Schwartz *et al.* [212] weighed the large combination of low-level multi-scale LBP and multi-scale Gabor

by tree-based partial least squares (PLS). Using PLS regression to weigh a combination of a large number of feature descriptors was proved to be robust in the unbalanced one-against-all classification scheme with highly biased class distributions with a single or very few samples in the positive class, and the application of a tree structure was efficient in reducing the computational cost of matching procedure.

- Two face recognition models were built with Gaussian covariance matrix on low-level descriptors: Chen *et al.* [213] utilized low-level LBP and measured similarity between image pairs with multivariate Gaussian covariance matrix. Similarity was calculated based on *joint* distribution of image pairs. Based on the paper, Cao *et al.* [214] used transfer Bayesian learning to train parameters on a big source-domain and added KL-divergence between the source and target domains to optimization objective. Both papers solved optimization via EM algorithm.

Sometimes preprocessing and postprocessing is applied before and after the transformation. Ren *et al.* [207] reduced the task to a two-class classification problem by performing CST before APCA. LBP deviated from Gaussian distribution for its non-negativity and simplex constraints while subspace transformation achieved optimal only under Gaussian assumption. CST was applied to LBP histograms to alleviate its non-Gaussian characteristics. Lei and Li [196] selected from relatively abundant features the most discriminative and suitable features for LDA by Adaboost and regression. The integration of regression into Adaboost was beneficial in that: the objective of regression was more consistent with subspace learning, and the solution could be obtained more efficiently by avoiding eigenvalue decomposition. Barkan *et al.* [191] down-regulated within-class covariance by Within Class Covariance Normalization (WCCN) metric learning after dimension reduction (WPCA, DM, PCA+LDA or DM+LDA). Rather than explicitly discarding dimensions, WCCN reduced the effect of within class directions by employing a normalization transform, and it could be done in either supervised or unsupervised way. Likewise, Ouamane *et al.* [32] applied Exponential Discriminant Analysis (EDA), a discriminative subspace method to address the small-sample-size problem ignored in LDA, followed by WCCN to downgrade the effect of direction of high intervariablility and enhance discrimination.

B. FEATURE SELECTION

Boosting is a frequently adopted supervised learning method that can be used for discriminative feature selection. We list some instances:

- Adaboost: Shan and Gritti [215] observed that in an arbitrary block division scheme, LBP-histogram (LBPH) bins contained useful information for expression recognition. Thus they proposed to learn discriminative LBP-histogram at bin level with Adaboost. For each Adaboost learner, the images of one expression were positive samples, while the images of all other expressions were negative samples. The weak classifier was designed to select the single LBPH bin which best separates the positive and negative examples. The selected LBPH bins reflected the significance of different facial regions, forming an informative compact facial representation. In a latter paper [189], the method was used for gender classification. As already seen in Section V-A, Lei and Li [196] preprocessed feature with Adaboost and regression. Liao *et al.* [193] applied Adaboost using the absolute difference between the same bin of two histograms as dissimilarity measure to select most effective from overcomplete uniform LBP features and construct face classifiers. Mendez-Vazquez *et al.* [216] selected processing configurations for input for face recognition in videos with AdaBoost based on χ^2 distances. Adaboost was also used to update the weight of each training sample such that misclassified instances were given a higher weight in the subsequent iteration [41].
- Boosting+Multi-Task Learning (MTL): In a face verification context, often a small number of training examples for each person are available for learning. If individual classifiers are learned for each celebrity, overfitting is inevitable. To overcome this issue, Wang *et al.* [217] presented Boosted MTL framework that *jointly* learned classifiers for multiple persons to overcome potential overfitting issues caused by lack of data.
- Boosting+Multiple Instance Learning (MIL): In multi-pose facial expression recognition context, proper patch with the default position should be located combined with rotation invariant features together to mitigate the misalignment problem. Hu *et al.* [166] used a boosting-based MIL approach to learn discriminative patterns on image patches and to alleviate the influence of image transformations due to misalignment.
- Boosting for ranking problems: Yao *et al.* [124] adopted RankBoost to select the most discriminative LGBP histograms. The algorithm combined the design of weak rankers and the selection of learner's parameters to minimize ranking loss, therefore more consistent with the objective of ranking. In the comparison of similarities of two images, the paper introduced constraints to force blocks corresponding to large similarities to have large outputs.
- Assembled boosting classifiers: Berg and Belhumeur's paper [218] was based on a set of images with face part locations and labels. After a specific identity-preserving alignment procedure, SIFT was extracted at manually located blocks and Adaboost was used to train linear classifiers for face verification. Classifiers were finally assembled.

Besides boosting, other discriminative histogram extraction methods based on various theories. We categorize them below.

- CMI-based: As have been discussed in Section V-A, Lei *et al.* [108] utilized CMI+LDA for global feature processing. The rationale behind CMI-based feature selection is: given selected features, the next feature should be selected to maximize the CMI.
- Multi-layer discriminant dominant feature selection: Guo *et al.* [219] invented a 3-layer model to estimate optimal pattern subset by considering robustness (L1), discriminability (L2) and representation capability (L3) across different classes simultaneously. L1 chose a subset of most frequently occurred patterns of a histogram, whereas L2 removed outlier within each class and fed to L3 for aggregation for all classes. L2 and L3 essentially equate FSC to represent patterns discriminately among training images belonging to the same class and across different classes, respectively.
- Random Forest (RF): Ghosal *et al.* [15] presented a framework that use RF to classify faces represented using Gabor wavelets. RF is an ensemble learning process which generates many classifiers and aggregates their results, which can be used to produce a measure of the importance of the variables. In a similar scenario with high-dimensional redundant LBP features, Connor and Roy [220] applied this RF framework to select important LBP features from extracted feature sequence to reduce the original feature space and speed up the classification process. Unlike the standard trees in which each node is created using the best split among all the variables, RFs split each node using the best among a subset of predictors chosen randomly at the node. This strategy seems to be contradictory. However, it performed relatively well compared to other classification techniques, including discriminant analysis, support vector machines (SVM), and neural networks, and is robust to overfitting.
- Mutual information maximization: To make LBP compact without redundancy, Jun *et al.* [221] proposed maximization of mutual information between LBP codes and class labels to select LBP code that retained discriminative information with reduced dimension. This approach to code selection iteratively selected the LBP codes which maximized the mutual information with respect to the class label, conditioned to codes previously selected.
- Similarity-based selection: Tran *et al.* [40], [167] extracted histograms based on similarity represented as mean vector and variance vector. The algorithm selected a subset of the extracted features that cause the smallest classification error.
- Kernel-based nonlinear discriminant analysis: Zhao *et al.* [222] proposed LBP based Kernel Fisher Discriminant Analysis (KFDA) by integrating LBP and KFDA method for face classifier. KFDA combined the merit of kernel based method and FLD: The nonlinear kernel method was adopted first to project the input data into an implicit feature space, and then FLD was performed in that space to produce nonlinear discriminant features of the input data. Kernel function was introduced by using Chi-square static distance and RBF as the inner product for KFDA classifier. KFDA was also employed in GaussianFace [132] to simplify the reformulation of DGPLVM, as already discussed in Section III-C.
- part-based/attribute-based: Inspired by their previous work [218], Berg and Belhumeur [79] learned two 10000-dimensional vectors for each face image pair, each component was the result of SVM classifier trained on base feature (gradhist, HOG, color histogram) for a particular part (image block). The workflow was more automatic compared to their early work: 16 face parts were located with landmark detection, and multi-scale weighted part blocks were generated automatically rather than manually. Kumar *et al.* [223] learned visual attributes and picked the best set of local features based on the attributes (gender, age, jaw shape, etc.) learned.
- Associate-Prediction model: Facial images in [224] were under significantly different settings. The paper offered a two-step scheme: first they block divided the faces into *components* and associated one input face with alike identities from the generic identity data set; then they predicted the appearance of one input face under the setting of another input face.

C. FEATURE FUSION METHODS

Rather than direct concatenation, fusion can be done on other levels. In the decision level fusion approach, separate classifiers are utilized to obtain scores based on local individual features, local decisions are then combined to obtain a final decision. It has certain advantages over feature-level fusion as features of different modalities may have various representation forms, and the fusion of decisions may be easier than the combination of features in a sensible way.

Decision level fusion is often a combination of output scores from classifiers. Fusion of LBP, Gabor and pixels scores were done in [199], and normalization was done as a postprocessing step. Taigman *et al.* [225] utilized the same local descriptors, fusing multiple LDA-based One-Shot Similarity scores. Similarly, Wolf *et al.* [226] applied local descriptors above plus Gabor features, they used a combination of Hellinger distance, one-shot distance, two-shot distance and ranking-based distances to gain high classification rate. Hu *et al.* [65] used Dense SIFT, LBP and multi-scale SIFT features and trained a deep neural network for metric learning; finally multiple features were fused at score level. Ding *et al.* [56] extracted facial textures from unoccluded blocks, and each block learned certain transformation dictionary for discriminative space projection. Cosine measure was utilized to calculate the similarity of each block pair and similarity scores of all blocks were fused by the sum rule. Local distances which were calculated by chi-square nearest neighbor classifiers regarding LBP and LPQ were fused in [192]. These scores were followed by

decision level fusion: a weighted combination of the results of the two, whose weights were employed based on mutual information.

Liu and Liu [96] had done multiple levels of fusion for different color components. DCT features extracted from multiscale Gabor representation for R image was fused via similarity score (determined by EFM) fusion. Multiscale LBP feature on C_r image was dimensionally reduced and concatenated. 3 component and multi-mask DCT extracted on Q image were fused by similarity score (determined via EFM). Finally, the representations in RC_rQ hybrid color space were fused at decision level using an empirically weighted sum rule.

Besides feature level concatenation and decision level fusion. Multiple-kernel Learning (MKL) lies somewhere between the two levels by estimating optimal convex combination of multiple kernels to train SVM. MKL appeared in [80] to combine LBP and HOG features for facial expression recognition, where ℓ_p-norm MKL algorithm was extended to a multiclass classification problem: rather than learning a joint kernel weight vector, respective kernel weight vector was learned for each binary classifier in the multiclass-SVM. In a similar manner, Chan *et al.* [227] combined kernels by direct addition without weight involved. Given LBP and LPQ features, each kernel function produced a square matrix in which each entry encoded a particular notion of similarity of one face to another. Once kernels were combined, Kernel Discriminant Analysis using Spectral Regression (SR-KDA) was applied for feature selection. In a later paper, Chan *et al.* [228] refined the system: LPQ was computed regionally with a component-based framework to maximize its insensitivity to misalignment; kernel fusion was compared with many fusion combinations; two geometric normalizations were used to combine scores of various image scales. Based on SR-KDA, Arashloo and Kittler [33] proposed CS-KDA (as mentioned in Section V-A) for component-based kernel fusion to construct a discriminative face descriptor out of multi-scale LBP and LPQ representations. Contrast to conventional approaches, CS-KDA recasted a multi-class classification problem into a set of two-class classification problems and its representation involving multiple shared kernel Fisher faces had only one class-specific kernel fisher face per class. Pinto *et al.* [24] blended 8 *V1-like* representation by the optimal combination of 6 kernels (element-wise squared difference, absolute-value difference, square-root absolute-value difference on *cropped* and *original* images respectively), resulting 48 kernels. The combination coefficients of kernels for classification were supervisedly tuned to optimal using the semi-infinite linear problem solver.

Finally, we would like to mention Hierarchical Ensemble Classifier (HEC) proposed by Su *et al.* [181]. In the paper, global feature was extracted from the whole face image by keeping low-frequency coefficients of Fourier transform, which encoded the holistic facial information, such as facial contour. Real and imaginary components were concatenated to form Global Fourier Feature Vector (GFFV). For local feature extraction, Gabor wavelets were exploited at every position and were spatially grouped into a number of feature vectors considering their biological relevance. This resulted in Local Gabor Feature Vector (LGFV). After dimensional reduction, GFFV and multiple LGFVs were used to train multiple component classifiers respectively. Finally, authors combined classifiers into one ensemble classifier by weighting. The proposed hierarchical ensemble method consisted of two layers of ensembles: the ensemble of all N local component classifiers (layer 1), and the ensemble of local classifier and global classifier (layer 2). Weights for layer 1 were trained with (interpersonal and intrapersonal) face pairs. For each image pair, a similarity vector could be obtained, with each similarity vector could be considered as a sample in the space of N dimensions. These similarity pairs were fed into FLD to get an optimal linear projection, resulting in the weight of LGFVs. Weights for layer-2 were trained similarly by the method.

A framework for combining multiple features by Struc [199] is a quite general illustration to the fusion pipeline. In detail, the first step used manually marked eye locations to normalize each color facial image. Then images were cropped based on bounding boxes of 3 sizes. Cropped images were then represented in the YCbCr color space, and luminance component was subject to a photometric normalization procedure to form a normalized version image. Color components of this image formed the basis for feature extraction, then Gabor and LBP features were computed and are subject to PCA. Finally, all feature vectors of the test image were matched against feature vectors in the gallery set to produce similarity scores. Scores were normalized and were combined using linear logistic regression (LLR). The basic framework is relatively clear: image patches of different color spaces, scales and local features were extracted; extracted features had gone under proper dimensional reduction and normalization before they were combined at score level based on LLR. A similar framework is observed in [203], though it applied feature-level fusion.

D. GLOBAL SPARSE CODING

This section is about spase coding face recognition model done directly on the whole face without block division. Many sparse face representation ideas are initially carried out globally without block division scheme, and there are still new ideas in global sparse coding. Therefore it is beneficial to summarize some basic practices.

Sparse coding is often combined with various machine learning theories to form a more robust reconstruction process. The kernel trick maps non-linear separable features into high dimensional feature space, in which features of the same type are easier grouped together and linear separable. In this case, the sparse representation for the signals are easily found, and the reconstruction error may be reduced as well. Motivated by this fact, Gao *et al.* [67], [229] proposed kernel sparse representation (KSR), a sparse rep-

resentation technique in a high dimensional feature space mapped by an implicit mapping function. KSR was also incorporated with spatial pyramid matching to achieve better performance. He *et al.* [230] developed fast face recognition based on sparse coding combined with Extreme Learning Machine (ELM). The paper extracted basis function from non-facial images, and then established single hidden layer feed-forward network to simulate sparse codes by ELM. The paper demonstrated the rationale of this common feature hypothesis. They obtained sparse coefficients from the universal image patches instead of directly from face image patches and then further processed them by means of ELM learning in single hidden layer feedforward networks trained with face images. Qian and Yang [231] introduced prior or statistical information learned from training data offline. The prior information matrices regarding reconstruction errors and representation coefficients formed a so-called General Regression and Representation Model (GRR) and were learned by using the generalized Tikhonov regularization, leave-one-out strategy in conjunction with K nearest neighbor algorithm. To make GRR robust to illumination, expression, and pose, authors extended the model to combine prior and specific information to give a weight to each image pixel with an iteratively reweighted method. To extend the single feature based SR model, Yuan *et al.* [232] generalized multi-task learning to a multi-task *joint* sparse representation model to combine pixel values and LBP feature for recognition. Each feature was then represented as a linear combination of the corresponding training features in a joint sparse way across all of the features. Finally, the classification decision was achieved according to the overall reconstruction error of the individual class. Zhang *et al.* [233] observed that multiple sparse representation vectors share sparsity patterns at class-level but not necessarily at atom-level. The paper collected facial images from *various view-points* and applied sparse coding with atom-level dictionaries: different images of the same person were jointly used to represent the person, with correlations among all views exploited and combined for discrimination. The proposed method allowed a flexible dictionary atom selection without requiring pose estimation. Wagner *et al.* [234] demonstrated that misalignment can be handled within the sparse coding framework. They sought the best transformation of the test image by minimizing alignment residuals, and they rejected invalid images according to sparsity concentration score.

There are approaches that applied supervision to dictionary learning scheme, and works below were inspired by the fact that traditional sparse overcomplete features extracted might not always be optimal in terms of discriminative power relative to the set of classifiers being considered. Pham and Venkatesh [235] formulated an optimization problem that combined the objective function of classification with representation error constrained by sparsity, forming a *supervised* joint representation and classification framework that sought for most discriminative sparse overcomplete encoding and optimal classifier parameters. K-SVD trained dictionary was further *iteratively* updated based on the outcome of a linear classifier, hence obtaining a dictionary that might be also suitable for classification in addition to having the representational power. Similarly, Zhang and Li [236] combined classification error into the objective function and incorporated labels *directly* (as opposed to iterative dictionary update procedure using feedback for the classification stage in [235]) in the K-SVD dictionary-learning stage to form Discriminative K-SVD. This dictionary-learning procedure found the globally optimal solution for the dictionary, its coefficients and classifier parameters based on label information simultaneously, thus had the potential of avoiding the local minima and its complexity was bounded by that of the K-SVD. A recent study called Supervised within-Class-similar Discriminative Dictionary Learning [237] combined the classification error term and the within-class similarity in the objective function of dictionary learning scheme for face recognition. The method was further extended by [238] to a multiple kernel fusion framework.

Like the initiative of collaborative representation, some papers focused on the optimizer itself. Timofte and Van Gool [239] proposed a classifier based on collaborative representation and regularized least squares. Differing from the original approach, authors observed that there was a difference in how helpful each training sample was in classification. Moreover, the features that described samples could discriminate among each other. Thus, two matrices were introduced to weigh each dimension of feature vectors and each training sample respectively. Then weights were considered for classification confidence to form Weighted Collaborative Representation Classifier (WCRC) to describe useful features. Wu *et al.* [240] extended WCRC to Learned Collaborative Representation Classifier by optimizing weights rather than intuitively determining them. Contrast to the former approach, Yang *et al.* [241] re-examined the sparse constraint to find its ℓ_1 optimizer offered not only sparsity but also closeness, where nonzero representation coefficients concentrated on the training samples with the same class label as the given test sample. ℓ_1 optimizer was more informative, as its objective function selected the support training samples to represent a given test sample with the minimal representation cost. By introducing the theory of global *neighborliness* and local *neighborliness* of quotient polytope associated with a dictionary, authors analyzed equivalence and rationale between ℓ_1 and ℓ_0 optimizers. Based on the closeness prior, the authors proposed two class ℓ_1 optimizer classifiers based on closeness and Lasso rule respectively. Peng *et al.* [242] advocated Locality-Constraint CR (LCCR) by introducing a novel objective function to code the training data and its nearest neighborhood to produce minimal reconstruction errors simultaneously, enforcing similar inputs to produce similar codes. LCCR aims to obtain a representation that could reconstruct the input with the minimal residual and simultaneously reconstruct the input and its neighborhood such that the codes are as similar as possible.

One notable merit of LCCR is that its objective function has an analytic solution and does not involve local minima. From a statistical viewpoint, Yang *et al.* [243] critiqued that fidelity term with ℓ_1- or ℓ_2-norm actually assumed that the coding residual follows Gaussian or Laplacian distribution, which might not hold with the occurrence of occlusions, corruptions and expression variations. Instead, they introduced an underlying distribution function and reformulated the problem into a sparsity-constrained Maximum Likelihood Estimation (MLE) problem and put forwards Robust Sparse Coding (RSC) scheme. Rather than determined the distribution explicitly, authors transformed the optimization problem into an iteratively reweighted sparse coding problem, and by iteratively computing the weights, the MLE solution of RSC could be solved efficiently. Results showed that RSC is robust to outliers. Later, Yang *et al.* [244] learned dictionary with the objective in the form of sparse coding combined with FSC-like terms; the dictionary was updated iteratively and the procedure was named Fisher Discrimination Dictionary Learning (FDDL). A first term enforced *representation fidelity*: each sub-dictionary had been represented well to samples from the corresponding class, but poorly to samples from other classes. A second term enforced *discriminativity* by minimizing within-class scatter while maximizing between-class scatter.

Codebook learning is crucial for reconstruction, thus is discussed in several papers. Jiang *et al.* [245] put forward Label Consistent K-SVD (LC-KSVD) by associating label information with each dictionary item and combining classification error in the objective function. LC-KSVD is a supervised algorithm that explicitly incorporates a *discriminative* sparse coding error criterion and an *optimal* classification performance criterion into the objective function and optimize it using the K-SVD algorithm. Discriminability is enforced in sparse codes during the dictionary learning process, as the learned dictionary is then both reconstructive and discriminative. To handle undersampled face recognition problems with only one or few non-occluded training images are available for each subject of interest, Deng *et al.* [139] proposed Extended SRC (ESRC) that built an intraclass dictionary with images collected from an external dataset that contained subjects not of interest to handle image variants. Then a given probe image could be reconstructed by using the single training sample which had the same class label with the query and the intra-class variance dictionary. Precisely, denote the SSPP training set as $A = [A_1, ..., A_n]$, intra-class dictionary as D, then ESRC reconstructs test sample as $y = A\alpha + D\beta + e$, where α and β are formulated as an optimization problem similar to SRC: $\min_{\alpha,\beta} \|\alpha\|_1 + \|\beta\|_1$, s.t. $\|y - A\alpha - D\beta\|_2 \leq \epsilon$ Later they proposed a ''prototype plus variation'' representation model [246], in which the dictionary was assembled by class centroids and sample-to-centroid differences. Authors showed that a sparse coding variant that represented test sample as a sparse linear combination of the class centroid and the differences to the class centroid, and this lead to high performance and robustness in uncontrolled environments. Unlike ESRC, Wei and Wang [247] advocated an optimization algorithm that *jointly* solved auxiliary dictionary learning and sparse representation. Similarly, Yang *et al.* [248] proposed to learn a compact dictionary with powerful variation representation ability *jointly* with an adaptive projection from the generic training set to the gallery set. By extracting from the generic training set a reference subset and a variation subset, the adaptive projection learning aimed to exploit the correlation between the reference subset and the gallery set, while the variation dictionary learning aimed to learn a compact dictionary with sparse bases from a big variation matrix (the projection of the intra-class variation of generic training set over the learned projection matrix). However, in a recent paper, Wei and Wang [249] observed limitations of the two papers: [247] viewed occlusion as the intra-class variation and demanded the information on occlusion of test images for learning intra-class dictionaries, while [248] treated occlusion as sparse errors and might be insufficient to represent the occlusion presented in real-world face images. Instead, they jointly solved the tasks of auxiliary dictionary learning and robust sparse coding in a unified optimization framework that learned intraclass dictionary without prior knowledge of occlusion by automatically disregarding unseen occlusions to make robust recognition. There are other approaches to perform face recognition across varying illumination and pose based on learning small sized class specific dictionaries. Patel *et al.* [250] presented a simultaneous sparse approximation: Given dictionaries learned with K-SVD for each class under sparsity constraint, test images were projected onto the span of atoms in each learned dictionary and the resulting residual vectors were used for classification. To recognize face under varying illumination and pose, image relighting based on pose-robust albedo estimation was used to generate frontal images under various lighting. Further, to reject nonface outliers, authors defined a rule based on the ratio between the norms of residual vectors of the first and second candidate. Ma *et al.* [251] proposed a discriminative *low-rank dictionary learning* for sparse coding, during which an objective function with sparse coefficients, class discrimination, and rank minimization was optimized. The integration of rank minimization into sparse representation for dictionary learning separated the sparse noises from the signals while simultaneously optimized the dictionary atoms to reconstruct the denoised signals. Thus, sparse noises in the training samples were corrected and the dictionary could be robustly optimized with an explicit discriminative goal, making the proposed model suitable for recognition. There are also approaches to address face recognition problems by considering non-Euclidean geometry. With the face described by LBP histogram, Harandi *et al.* [252] explored sparse dictionary learning over Grassmann manifolds. A method for learning a Grassmann dictionary extrinsically was introduced and Grassmann manifolds were embedded into the space of symmetric matrices by a diffeomorphism that preserved Grassmann projection distance.

VI. SUMMARY AND EXAMPLES OF FACE FEATURE EXTRACTION

Here we made a summary of face feature extraction based on local features and filters, codebook and encoder, and holistic encoding. We selectively list some of the feature extraction procedures in Table 4.

We give two typical examples to further illustrate the idea of face feature extraction. One framework is the learning-based encoder adopted by Cao *et al.* [45]. The major novelty of the approach lies in the *encoder* design: different sampling method and an RP-tree to quantize the feature vector to discrete codes. Image patches were extracted on DoG filtered image, next an LBP-like sampling procedure was used, then machine learning approach was used to encode features, and finally the coded image was split and concatenated. Proper dimensional reduction and normalization were used between building blocks. Please note that [44] adopted an extremely similar framework, though the CITP tree was used for coupled encoding photo and sketch simultaneously.

Another illustration would be the DFD-based face *representation* of Lei *et al.* [36]. It is essentially the same as that of [45]: learned DFD for each face region was concatenated to form a histogram. However, unlike the previous approach, initiative of learning procedure of [36] is quite different. It operated on Patch Difference Matrix (PDM) as local features, and then extracted *discriminant* features for clustering and representation basis. Images were projected to a subspace determined by *image filter vectors* and *soft sampling vectors*. Both vectors were learned through optimization according to FSC: they minimized within-class scatter and maximized between-class to learn discriminative filters to extract local features as well as to find optimal weights for the contribution of neighboring pixels in computing the descriptor. Finally, image pixels were finally labeled according to the dominant patterns found in the K-means clustering phase. A similar framework is observed in [22] and [38], where the latter projected image by a learned filter and then encoded the filtered image via LBP.

VII. DEEP LEARNING (DL)

Face recognition on regular databases with co-operating individuals in a constrained environment is challenged as our applications progress to unconstrained scenarios with uncooperative subjects, like forensic investigations of video surveillance. Persons to be identified may be under various settings and the extraction of discriminative patterns would fail because of the unpredictability.

In this sense, face recognition tasks are rather like a complex learning problem without much prior knowledge and common patterns, and according to [253], that is what deep learning algorithms are for. The paper stated that prior knowledge is rather important in generalizing a pattern for a database. Without proper prior, a huge database is required in order to obtain generalization. We wish to find an efficient algorithm with minimum requirement for prior, labeled training data and human interference. Traditional non-parametric methods (e.g. kernel methods), given their ability to use convex optimization, suffer inefficiency in representing a family of functions in *high dimension*. Comparatively, deep architectures outperform shallow ones in that they offer better *scaling* properties. Furthermore, given our ignorance of priors, we would unavoidably seek non-convex loss functions.

Traditional methods lack trade-off between 'breadth' and 'depth'. Their outputs tend to be very local (in feature space), thus is dominated by patterns of the nearest neighbor. This result in the *curse of dimensionality*. If we use hyperplanes to encode our representation for classification, a number of subregions obtained is exponential to the number of hyperplanes used. Traditional non-parametric methods like kernel methods, nearest neighbors, and mixture models made a very weak assumption on the function to be learned, so the representation power grew with training data. It is proved in [253] that for kernel methods, we have to obtain at least exponential amount of training data. For our recognition problems, facial image transforms like shifting or rotation, tend to result in a non-linear manifold with high curvature.

We seek non-local learning approaches. Some non-local models like polynomial fitting generalize poorly at unseen places. Deep learning, however, learns non-locally by a series of nonlinear transformation. We can base each layer of our deep model on kernel methods, and by multi-layer architecture, high-level structural features like eyes and nose can be generalized. This procedure is fully automatic without human intervention. Once we fall short of labeled data, we could utilize unsupervised *pre-training* techniques to learn high-level representation, and apply learned representation for classification. Matching high-level features are efficient in that we would need less labeled samples, as most information is acquired in an unsupervised fashion. Thus the key to the success of DL lies in its ability to learn better representations, mostly from unlabeled data. Additionally, non-convex error functions in high dimensional spaces tend to result in the proliferation of saddle points [254], which alleviate local minima convergence problem.

In previous sections, the core feature extraction procedure is to extract features locally and do feature transformation to project them to a subspace to form a global representation. DL can be seen as a cascade of multi-layer convolution and nonlinear transformations. Its parameters are obtained by optimizing the loss function. This layer-wise formulation learns multiple levels of abstraction from local to global, thus forming a hierarchical feature representation. Discovering intermediate representations can be beneficial to tasks like domain adaptation [255], [256], as learning dependencies in latent variable space is easier than the raw input. It could also be beneficial to multitask learning, for instance in [257] facial expression attributes help detection of landmarks and pose. Those advanced and shareable features could also be beneficial in transfer learning, as lower-layers are similar to Gabor features and not specific to particular dataset or task. In [258], it is found that initializing a network with transferred features could improve generalization performance even after

TABLE 4. Various feature extraction approach.

Local features and filters	Encoding	Representation and holistic encoding	Reference
Pixels	Decision tree (unsupervised)	PCA+LDA (supervised) and identity factor analysis matching	[157]
Pixels	DT-LBP (supervised decision tree)	Concatenation of histograms	[155]
Pixels	ICA (unsupervised) for each region	Binarization by thresholding (unsupervised), concatenation	[102]
PDV	Codebook(generated by K-means)	FSC weighted histogram intersection (supervised)	[36]
DoG+LBP-like low-level feature	K-means/PCA tree/ random projection tree (unsupervised)	PCA + normalization	[45]
Filtered image patch	Histogram based on K-means	N/A	[11], [16]
LBP	Riemannian VLAD	N/A	[130]
Variants of LBP	Histograms	Histograms concatenation+PCA+EFM	[161]
Variants of LBP or SIFT	N/A	WPCA + Unsupervised WCCN (unupervised) PCA+LDA+WCCN (supervised)	[191]
LPQ	Histograms	EDA+WCCN+score fusion	[32]
binarized pixel difference	Select code by maximizing mutual information (unsupervised)	PCA	[92]
MLBP	N/A	Feature selection(Adbaboost+regression)+LDA	[196]
Various transformations	LBP	Various subspace representations (unsupervised: PCA, UDP; supervised: KCFA, KDA, LDA)	[120]
Joint LBP feature of image pairs	N/A	DGPLVM (supervised)	[132]
LQP or G-LQP	Histogram	WPCA	[107]
GV-LBP	Weighted histogram	CMI+LDA	[108]
Gradient image	LBP+Coupled encoding via CITP tree	Concatenation of histograms+PCA+LDA	[44]
PDV	GMM (unsupervised)+Fisher score (unsupervised)	Normalization	[37]
Fused features of Gabor and LBP	N/A	KDCV	[203]
Multiscale LBP and LPQ	Histograms	CS-KDA+kernel fusion	[33]
Cascade of PCA filtered image	Binarization, indexing and pooling	Concatenation	[169]
Image patches	K-means+BoW (unsupervised)	N/A	[128]
SIFT and SURF	VLAD	N/A	[69]
Multiscale Gaussian derivatives with spatial locations	Random projection tree (unsupervised)	N/A	[158]
LWHT complex images	Phase Magnitude Histograms	Block-based WPCA	[84]
RDF generative model	FV	N/A	[74]
Local feature (coordinates, intensity, derivative)	FV	N/A	[202]
PDV	Binarized feature mapping (unsupervised)	Clustering and pooling to form histograms	[34]
PDV	SLBFLE	N/A	[35]
DCP on overlapped patches and discriminative projection	N/A	Score fusion	[56]
Gabor and intensity	N/A	Multi-kernel learning	[24]
Gabor	N/A	EFM	[14]
Gabor	Histogram based on K-means	Histogram weighted by FSC	[22]
Gabor	Quantization, forming spatial histogram	Histogram weighted by FSC	[21]
SIFT	Randomized trees	N/A	[153]
Dense SIFT	FV	Discriminative metric learning	[68]
Dense SIFT	FV with foreground/background selector	N/A	[131]
SIFT,LBP fused	PEP	N/A	[71]

fine-tuning on a new task or dataset. This shareability enables us to use deep networks trained for face identification to do facial expression recognition. We will discover more constructive properties of DL in this section.

Many papers in the computer vision society are based on a DL framework (or Deep Neural Network, DNN) ever since the introduction of Alexnet [259]. It gains popularity due to its stunning performance at image classification tasks, though it receives notoriety due to its intricacies. We review papers which utilize DL for face recognition. There are two major frameworks for DL: the first is Deep Belief Networks (DBN), which consist of Restricted Boltzmann Machines (RBM); the second is Convolutional Neural Networks (CNN).

A. LOCAL FEATURES AND INTEGRATING SPATIAL AND SCALE INFORMATION

Low-level features are used in several papers. Huang *et al.* [260] used LBP as well as pixels as local representation. They demonstrated by applying DL to LBP its potential to capture complementary higher-order statistics of hand-crafted descriptors. Yi *et al.* [261] first extracted Gabor features at localized facial points, as Gabor features were considered to have a strong discriminative ability and were robust to variation. Liao *et al.* [262] extracted 3 kinds of features for comparison: LBP, HOG, LBP+HOG of 31 scales, and the paper provided PCA of different dimensions to local features.

Non-overlapping block partition was observed in [263], where features were aggregated by concatenation. In [260], 9 overlapping regions was operated on LFW database, and to attain multi-scale features, the images were cropped to patches of 3 different sizes. In some papers, manually selected regions of different scales were used [264], [265]. Note that in [265], 400 face blocks were used, but the number was reduced to 25 with the help of a greedy algorithm to select most effective and complementary feature vectors.

In [266], 3-scale patches around landmarks and particular patches of the global region was obtained. Fan *et al.* [267] achieved multi-scale by elaborating Pyramid CNN. A recent paper [268] proposed to extract the local adaptive convolution features from the local regions of the face image motivated by the success of local features and the deep convolution features. Deep convolution features of most informative parts of the face and densely sampled points were used to represent the face. Joint and collaborative representation of all convolution features was introduced to exploit distinctiveness and commonality of various local regions. Representation coefficients of different regions were required to be similar because these local regions came from the same query image and proposed a joint collaborative representation model to effectively fuse the local deep feature representations in different locations, and the paper specifically addressed SSPP problem.

Pooling is a procedure for gaining spatial invariance and it recurs in various DL papers. Max pooling is observed in [257], [260], [265], [266], and [269]. HMAX (sum pooling) was used in [262]. 2 level average pooling was adopted in [264], which would be discussed later in VII-B. K-means was used to learn codebooks to pool features in [263]. Roychowdhury *et al.* [270] put forward 'score pooling' and 'feature pooling', where max pooling was operated on SVM scores and features respectively.

B. FRAMEWORK

DBN is a generative graphical model composed of multiple layers of hidden variables with connection between layers. In a heterogeneous facial recognition scheme [261], local RBMs were trained with Gabor features, and finally PCA was performed. In the network, the task of Gabor feature extraction was to extract discriminant and robust features for each modality, and RBM tried to build the relationship between two modalities. Nair and Hinton [271] replaced binary hidden units in RBM by *noisy rectified linear units* (NReLUs), where the value of a hidden unit was given by the rectified output of the activation with added Gaussian noise. RBMs with NReLUs were pre-trained generatively, and then trained discriminatively using backpropagation. The network used for face feature extraction contained one hidden layer of NReLUs pre-trained as an RBM. Then cosine distance between network output of two facial images was computed. The network was translation equivariant and scale equivariant.

Some *multi-view* face recognition frameworks were inspired by auto-encoders. Zhang *et al.* [272] proposed face verification procedure based on single-hidden-layer neural network-based with sparse constraint. This *shallow* network was guided by multiple *random* faces as target values for multiple encoders. The rationale behind random faces as target was our need for identity representation. Introducing multiple random faces enabled learning of multiple encoders which randomly encoded private or common attributes to the identity feature and artificially produced many random shared structures between two identities, thus enhanced discriminative power of proposed identity feature. By enforcing the target values to be unique for input faces over different poses, the learned high-level feature that was represented by the neurons in the hidden layer was pose free and was only relevant to the identity information. Wang *et al.* [273] put forward Deeply Coupled Auto-encoder Networks (DCAN) to tackle cross-view face recognition problem. DCAN was based on two DNN (one for each view) coupled with each other in every corresponding layer. The two different non-linear DNN avoided potential inadequate representation capacity of simpler models. Each DCAN structure was developed by stacking locally consistent and discriminative coupled auto-encoders (trained with maximum margin criterion as a single layer component), resulting in a narrowing gap between two views and gradually improved shared features in the common space. Kan *et al.* [274] built a Stacked Progressive auto-encoders (SPAE) model by stacking several shallow auto-encoders to a DNN. To mitigate the highly complex non-linear transform that directly transforms non-frontal faces to frontal, authors decomposed the problem to multiple tractable (less non-linear) phases by progressively narrowing down pose variations to zero. Each auto-encoder converted input face images at large poses to virtual views at smaller poses as well as keeping smaller poses unchanged.

CNN is popular in the computer vision field. Usually, it consists of a various combination of convolutional layers with Rectified Linear Units (ReLU) activation function, pooling layers and fully connected layers (FC). Since FCs contains most of the parameters and prone to overfitting, *dropout* method is introduced to prevent overfitting. Finally, the loss layer is where a form of the loss function is applied for a specified task.

Cox and Pinto [275] combined multiple complementary representations via kernel fusion. Firstly, they offered two multi-layer feature extraction frameworks that closely resembled a hierarchy of 2 (HT-L2) and hierarchy of 3 (HT-L3) CNN respectively. Each layer included: a linear filter which initiated randomly and learned supervisedly; activation function that consisted of threshold and saturation function; pooling layer that did spatially downsampling; normalization process. They later blended the one layer V1-like features along with multilayer HT-L2 and HT-L3 with kernel fusion. Sun *et al.* [266] built multi-scale CNN to learn a set of high-level features with image patches as input, and its *final representation* (Deep-ID layer) was the hidden layer before softmax. Deep-ID layer was directly and fully connected with the third and fourth convolutional layers (after maxpooling) of the network so that it saw multi-scale features. Authors argued that this was critical to robust feature generation due to the successive down-sampling along the cascade that caused fourth convolutional layer contained too few neurons and becomes the bottleneck for information propagation. Zhou *et al.* [276] explored traditional sophisticated methods (Joint Bayesian, clustering, etc.) and observed that as training data increased, little gain was obtained by these methods.

Instead, they simply established a ten layer naive CNN network, whose last layer was softmax layer for supervised learning. Like [266], the *final representation* was the hidden layer before softmax followed by PCA model. The similarity between two images was measured through a simple ℓ_2 norm. Taigman *et al.* [269] demonstrated that coupling a 3D model-based alignment with large capacity feedforward models images was beneficial to effective representation. In their work, images were aligned with 3D face and then represented with CNN. Recognition accuracies against various traditional methods demonstrated model effectiveness. Contrast to [269], Zhu *et al.* [277] trained CNN to recover canonical view of face without 3D information. Authors devised a facial measurement for frontal view face images by combining the rank and symmetry of matrix. The framework used such terms to select frontal face image for each person, and CNN learned the regression from various view to canonical view. Then, image patches extracted according to 5 landmarks from frontal face image were used to train another CNN. Features (outputs) of the network followed by PCA was used for verification. Bilinear CNN (BCNN) was applied to face identification in [270]. BCNN consisted of two CNNs whose outputs are multiplied at each location of the image. Advantages of BCNN model are: it can be trained using only image labels without requiring ground-truth part-annotations; direct end-to-end training is applicable, which allows initializing generic networks with pre-trained networks (like ImageNet) and then fine-tune them on face images. Multi-scale feature sharing representation was put forward in Fan *et al.* [267] and was named Pyramid CNN. Its motivation was to accelerate the training of deep neural networks and to take advantage of the multi-scale structure of the face. Pyramid CNN consists of multiple levels of networks. These networks have different depths and input sizes, and they share some of the layers. The first layer is shared across all levels, while the second layer is shared by networks from the second level. This sharing scheme is repeated and thus the size of the network that is actually trained does not grow as the level increases. Further, in contrast to train the deep network directly, the procedure can be decomposed into training several small networks. Hierarchical feature representation was the core of [263]. For each layer, authors firstly *jointly* unsupervisedly learned multiple yet related feature projection matrices for different face regions simultaneously and neighborhood blocks *sparse* share codebooks. By saying *joint*, authors learned feature projection (or weighting) matrices W from different face regions *jointly*, as regions, though separated, usually share some related information in feature representation. Secondly, features of each patch were then projected with W and then converted to discrete codes with dictionaries generated by K-means. This joint feature learning model was stacked into a deep architecture to exploit position-specific discriminative information for face representation. WPCA was used within layer: output feature of layer 1 was WPCAed and then fed to layer 2. The network harvested decent performance in hierarchically extracting position-specific discriminative information and jointly learned multiple related features for different face regions.

Similar to canonical-view face recovery with CNN [277], some authors altered CNN to tackle various problems like landmark detection and pose generation. Within the framework of CNN, Zhang *et al.* [257] investigated optimizing facial landmark detection with related auxiliary tasks: head pose estimation, gender classification, age estimation, facial expression recognition and facial attribute inference. The proposed Tasks-Constrained Deep Convolutional Network, a 5 layer CNN with MTL optimization objective, allowed errors of subtly related tasks to be back-propagated in deep hidden layers for constructing a shared representation to be relevant to the main task. Thus it did not only estimate landmarks but poses and attributes as well. Liao *et al.* [262] elaborated a detection, alignment, recognition pipeline by storing templates (transformations of training images), and fed into a feedforward hierarchical network (like HMAX hierarchical computational model of the visual cortex). The hierarchical architecture consisted of a repeated biologically-plausible V1-inspired modules. At the first layer they obtained low-level features; the second layer extracted dense overlapped *windows* and *templates* for convolution and pooling; the third layer computed dot product for person identification. To alleviate the computational bottleneck in the second layer, the system was approximated by local sensitive hashing-based voting for templates and windows. Jung *et al.* [278] proposed a new DNN based on multi-task learning that rotated an arbitrary pose face to a target pose while preserving identity. This DNN took a face image as well as a Remote Code, which represented the target pose code corresponding to the output image. This DNN bore similar structure with CNN, however, in the first part it did not share filter weights due to the presence of Remote Code and in the second part, fully connected layers were applied within each layer to contain pose information into features while preserving identity. To further improve identity-preserving ability, authors introduced an auxiliary DNN and an auxiliary task: pose-robust features were required not only to reconstruct face image under target pose but also to recover original input image. Recognition rates for various poses were reported. Ding *et al.* [279] proposed a composition of a set of CNNs and stacked auto-encoder to jointly learn face representation using several cropped facial blocks. CNNs extracted complementary facial features from multimodal facial blocks and they were then concatenated to form a high-dimensional feature vector, whose dimension is compressed by sparse auto-encoder. Following [279], a set of DCNNs are used to perform data fusion on complementary facial features and multimodal data extracted [280], [281].

The idea of deep feature learning guided by *both identification and verification signals* were initiated in Sun *et al.* [265], who utilized the two supervisory signals in the 3rd and 4th convolutional layer. Feature output (DeepID2) was 160 dimension for each of the 25 face blocks and output of image blocks was concatenated and applied PCA to reduce

dimension. Identification signal classified each face image into one of the identities and was achieved with a softmax layer. Face verification signal encouraged features extracted from faces of the same identity to be similar and it directly regularized the feature vector and could effectively reduce the intra-personal variations. The network was trained to minimize the cross-entropy loss regarding identification signals, and ℓ_1-norm or ℓ_2-norm or cosine measure regarding verification signals. Methods to balance the identification and verification signals is also discussed in the paper. In a later paper [282], larger DeepID2+ features of 512 dimension were learned. The network was larger with more training data and supervisory signal was used in each layer. Please note that in this system, neural activations were moderately sparse (about half of the neurons are activated), highly selective to identities and identity-related attributes (gender, race, age, hair color, etc.) and much more robust to occlusions.

A novel system called FaceNet was introduced in [283] that *directly* learned a mapping from face images to compact Euclidean spaces. Distances were directly a measure of face similarity. In the system, CNN was followed by normalization and triplet loss layer. The method trained CNN with triplets of face patches directly to optimize the embedding without intermediate representation layer, which might be a bottleneck for performance and required processing like PCA. The network strived to seek for an embedding from an image into a feature space, such that the squared distance was small between all faces of the same identity while large between a pair of different identities. Triplet loss enforced a margin between each pair of faces from one person to all other faces and experiments showed the system has high representation efficiency with only 128-bytes per face.

VGGnet [284] and GoogLeNet [285] are results of designing a really *deep* network. These Network-in-Network (NIN) [286] based approaches are powerful in learning robust patch representation to capture the structure, and experiments proved its power. In light of these, Sun *et al.* built two *DeepID3* networks [287] (with 10+ layers). They trained one network with inception layers used in GoogLeNet architecture and another one with stacked convolution layers. Finally, outputs from two networks were concatenated. Inherited from [282], the network included unshared weights in the last few feature extraction layers and supervisory signals are added to early layers in a similar manner.

Even though particular convolution layers have been used to reduce CNN parameters, problems still arise as these really deep networks could be hard to train: it cost a huge amount of time, space and its convergence is a challenge. A recent study [288] alleviated this issue by put forward a *lightened* CNN framework. Authors substituted ReLU-based activation function with Max-Feature-Map (MFM), a *maximum* between two convolution feature map candidate nodes. MFM can be seen as sparse connections between layers to achieve variable selection dimension reduction, thus compact representation was obtained. The paper provided two different architectures. Model A was consists of 4 convolution layers, 4 max-pooling layers and 2 FCs. Model B was based on 5 convolution layers, 4 NIN layers, 4 max-pooling layers and 2 FCs. Training time for Model A and B was merely 1/8 compared to VGGnet and parameter storage space was about 1/18, whereas its face recognition performance on LFW and Youtube Face dataset outperformed VGGnet.

TABLE 5. Various DL framework.

Framework	Details and Reference
DBN	shared representation (local RBMs are trained with Gabor features) + PCA [261] RBMs with NReLUs[271]
CNN	output layer directly connected with the penultimate and antepenultimate layer [266] bilinear CNN [270] 5-layer CNN+multitasking [257] frontal alignment by 3D face + 7 layer CNN [269] CNN with feature sharing[267] joint feature projection learning[263] supervisory signals involved [265], [282], [287] two lightened CNN models with MFM [288], inception layer and stacked convolution [287] one CNN for canonical view recovery and another CNN for feature extraction [277] CNNs combined with stacked auto-encoder[279]
CRBM	self-taught trained CRBM with altered energy function[260] supervised CRBM[289]
Hybrid CNN-RBM	joint local relational feature extraction[264]

Finally, Convolutional RBM (CRBM) is a hierarchical generative model that scales to full-sized images. In CRBM, local weights are shared among all locations like CNN does, which makes it possible to use high-resolution images as input. CRBM also introduces probabilistic max-pooling, allowing higher-layer units to cover larger areas of input probabilistically. Thus CRBM could have local translation invariance and still allows top-down and bottom-up inference in the model. Huang *et al.* [260] built convolutional DBN by self-taught learning to learn complementary representations of LBP and two layers of CRBMs were stacked to form a DBN. The model had used an alter energy function to form local CRBM, and PCA was computed on the represented feature. Finally, score level fusion of raw pixels and LBP features was done using a linear SVM. Alternatively, Zhu *et al.* [289] supervisedly learned local CRBM, the output of final layer formed face-identity preserving (FIP) feature. Firstly, input images were encoded through feature extraction layers, which had three locally connected layers and two pooling layers stacked alternately, and each layer captured face features at a different scale. Each locally connected layer output 32 feature maps, where each map had responses inside and outside of the face region to capture face structures and pose information respectively. Secondly, the FIP features recovered the face image in the canonical view using a fully-connected reconstruction layer. To train the parameters of the network, authors proposed to firstly initialize the parameters based on the least square dictionary learning and then updated them by back-propagating the summed squared reconstruction error between the reconstructed image and the ground truth. Experiments showed that the network was robust to illumination and pose and could reconstruct face to canonical view. Contrast to approaches in previous sections, in this paper recovering canonical-view face images did not either rely on a 3D face model nor depend on prior information on pose or lighting condition. This further proves that deep representation could model multiple complex transformations as well as disentangle hidden factors through feature extraction

TABLE 6. Verification performance on LFW.

Accuracy(%)(Protocol)	Details
73.4 ± 0.4(URLF)[37]	Preprocessing: aligned; resized to 50×40; divided to 5×4 blocks Descriptor: Local Higher-order Statistics w.r.t. 128 GMMs ($5 \times 4 \times 128 \times 8 = 20480$ float) Classifier: Nearest neighbor with thresholding (threshold trained supervisedly)
93.3 ± 1.3(IRLF)[218]	Descriptor: SIFT extracted at several points and scales Classifier: Tom-vs-Pete classifier trained with Adaboost; attributes classifier outputs appended
84.5 ± 0.5(URWL)[45]	Preprocessing: landmark detection Descriptor and classifier: combination of 4 Learning-based descriptor via ℓ_2 similarity scores and SVM; applied a set of linear SVM to perform component-level, pose-adaptive matching
75.4 ± 0.7(IRLF)[117]	Preprocessing: aligned; cropped to 110×116 pixels; divided to 10×10 non-overlapping blocks Descriptor: unsigned POEM representation with 3 bins (7 discretized orientations each); accumulated over all pixels of a 7×7 patch with 59 uniform patterns($10 \times 10 \times 3 \times 59 = 17700$uint) Classifier: flipped images of each pair vertically and took the smaller of the histogram distances as measure
86.2 ± 0.5(U)[107]	Preprocessing: cropped to 80×150 pixels, divided to 10×10-sized blocks Descriptor: intensity LQP with 150 words codebooks ($150 \times 2 \times 8 \times 15 \times 2 = 72000$uint); WPCA of dimension 2000 Classifier: nearest neighbor with cosine measure
87.8 ± 0.5(IRLF)[17]	Descriptor: pose adaptive feature of 176 points, each point represented by Gabor feature of 5 scales and 8 orientations ($176 \times 5 \times 8 = 7040$ complex value); PCA Classifier: nearest neighbor (rank-1)
88.6 ± 0.4(U) 91.1 ± 0.6(IRNO)[191] 92.1 ± 0.5(URNO)	Preprocessing: aligned; cropped to 150×80 Descriptor: LBP(7080int), TPLBP(40887int), Over-Complete LBP(9216int), SIFT(3456float), Scattering(96520float);WPCA to 500 dimension, then LDA (if applicable) to 100, 100, 100, 30, 70 respectively; applied WCCN to dimensionality reduced descriptors Classifier:similarity thresholding to Equal Error Rate point and Verification Rate point; or SVM
84.0 ± 0.4(ULF) [36]	Preprocessing:divided to 7×7 non-overlapping regions Descriptor: DFD of scale 5 with 1024 dominant patterns ($1024 \times 49 = 50176$float), WPCA reduced to 1100 dimension Classifier: nearest neighbor (rank-1)
88.7(IRNO)[157]	Preprocessing: aligned; cropped to 200×150 Descriptor: Maximum Entropy Feature Descriptor based on MLBP and SIFT at 4 scales ($M \times N \times 4 \times (256 + 128)$float, $M \times N$ is image block division number) Classifier: nearest neighbor (rank-1)
90.9(U)[34]	Preprocessing: 3 versions of alignment and cropping Descriptor: Compact Binary Face Descriptor to the 3 versions of image with local region 8×8 and codebook size 500 (32000int); WPCA to 700 dimension Classifier: average of the 3 descriptors; nearest neighbor (rank-1)
92.6 ± 1.1(IRLF)[34]	Preprocessing: 3 versions of alignment and cropping Descriptor: Compact Binary Face Descriptor to the 3 versions of image with local region 8×8 and codebook size 500 (32000int); WPCA to 700 dimension Classifier: average of the 3 descriptors and SVM score level fusion
93.8 ± 1.3(URLF)[34]	Preprocessing: 3 versions of alignment and cropping Descriptor: Compact Binary Face Descriptor to the 3 versions of image with local region 8×8 and codebook size 500 (32000int); WPCA to 700 dimension Classifier: average of the 3 descriptors and SVM score level fusion
85.6(U)[35]	Preprocessing: cropped to 128×128, divided to 8×8 blocks Descriptor: PDV extracted with 3 different radius, SLBFLE with dictionary size set to 600 ($8 \times 8 \times 600 = 38400$int), WPCA to 500 dimension Classifier: cosine similarity
81.7 ± 1.8(IRNO)[71]	Preprocessing: no alignment; cropped to 150×150; 3 scale Gaussian image pyramid Descriptor: PEP (difference vectors of SIFT and LBP w.r.t. UBM-GMM of 1024 Gaussians; augmented with coordinates of patch center) Classifier: SVM
91.1 ± 1.5(IRNO)[73]	Preprocessing: cropped to 150×150 Descriptor: 3 layer hierarchical-PEP; PCA reduced to 200 dimensions; layers fusion Classifier: SVM
87.8 ± 0.6(IRLF)[260]	Preprocessing: 3 versions of cropping and resizing; Descriptor: binary CRBM on LBP and perform PCA (500 dimension); combined the DL feature with hand-crafted features Classifier: cosine similarity metric learning + SVM

TABLE 6. Verification performance on LFW.

91.75(URNO) 92.52(URWL)[264]	Preprocessing: 3-point alignment Descriptor and classifier: trained CNN with raw pixels; pooled output of 12 groups of 5 layered CNNs as mid-level descriptor; RBM with input layer size 12, hidden layer size 14, output layer size 2
93.2(U) 93.2(URNO) 95.2(URWL)[29]	Preprocessing: 27-point alignment Descriptor: 27 landmark-based dense (16 girds) 5-scale sampling; high-dimensional LBP ($59 \times 16 \times 27 \times 5 = 127440$ int), PCA reduced to 400(U/URNO) or 2000(URWL) dimension; further compressed by rotated sparse regression with sparsity 0.95 Classifier: cosine similarity
90.9(URNO) 92.4(URWL)[213]	Combined score of joint Bayesian model based on LBP, SIFT, TPLBP and FPLBP
96.3 ± 1.1(URWL)[214]	Descriptor: multi-scale LBP centered at dense facial landmarks (dimensional > 100000), PCAed to 2000 Classifier: joint Bayesian model with transfer learning (source domain:WDRef, target domain: LFW)
97.2 ± 0.3(IRWL) 97.4 ± 0.3(URWL)[269]	Preprocessing: frontalization by 3D face modeling Descriptor and classifier: Ensemble of 3 CNNs based on different types of inputs, each with > 120 million parameters using several locally connected layers without weight sharing; CNN inputs were: 3D aligned RGB,gray-level image and gradient, 2D aligned RGB images
91.8 ± 0.6(URLF)[56]	Preprocessing: pose normalization based on 5 points; occlusion detection Descriptor: extracted pair-wise DCP features from un-occluded patches only; reduced dimension by PCA; learned transformation dictionaries by MtFTL Classifier: metric learning based
96.1(URWL, JB) 94.3(URWL, NN)[266]	Preprocessing: aligned with 5 landmarks Descriptor: extracted features from 10 regions, 3 scales and RGB/gray channels; trained 60 CNNs, each of which extract two 160 dimension DeepID (altogether $10 \times 3 \times 2 \times 2 \times 160 = 19200$dimension) Classifier: JB or NN for face verification
99.2 ± 0.1(URWL)[265]	Preprocessing: aligned based on 21 landmarks; cropped 400 patches with variations Descriptor: 400 DeepID2 vectors extracted by 200 CNNs, each of which was trained to extract two 160 dimension DeepID2 vectors; selected 25 effective ones (altogether $25 \times 160 = 4000$ dimension); further compressed by PCA to 150 Classifier: JB model on selected features of 7 iterations; classify with score fusion and SVM
99.5 ± 0.1(URWL)[282]	Preprocessing: same as [265] Descriptor: similar to DeepID2 network, but extracted 512 dimension DeepID2+ vectors from each of the 4convolutional layers (altogether $512 \times 4 = 2048$ dimensions) Classifier: JB model trained on 2000 facial data not used previously
93.0 ± 1.0(URWL) 87.5 ± 1.5(IRWL)[68]	Preprocessing: extracted a 160×125 face region, took horizontal reflections of compared images and average the distances Descriptor: computed dense PCA-ed SIFT of 64 dimensions, appended coordinates of locality (2-dimension), clustered with 512 GMMs and encoded with FV ($2 \times (64 + 2) \times 512 = 67584$double) Classifier: joint metric similarity learning on 256 dimensions
85.6 ± 1.5(IRLF)[131]	Preprocessing: extracted a 150×150 face region Descriptor: computed dense PCA-ed SIFT of 64 dimensions, appended coordinates of locality (2-dimension), clustered with 512 GMMs, applied foreground/background selector for descriptor and Gaussian codebooks and encoded with FV ($2 \times (64 + 2) \times 512 = 67584$double) Classifier: nearest neighbor (rank-1)
93.1 ± 0.4(IRWL)[79]	Preprocessing: aligned w.r.t. part locations, cropped to 64×128 Descriptor and classifier: took 2-scale grids (8 and 16×16), then extracted gradhist and color histogram feature within each grid and concatenated them.SVM is trained to separate two classes and score from SVM was the score of that part-based feature (two 10000-dimensional vectors). This feature from training pairs were used to train a same-vs-different classifier that makes verification decision.
99.5 ± 0.4(URWL)[276]	Descriptor: a 10-layer network, where output of the penultimate layer followed by PCA is was used as face representation Classification: ℓ_2 norm measure
96.5 ± 0.3(URWL)[276]	Descriptor: recovered canonical view and extracted 5 landmarks; image patches were cropped based on landmarks; each patch pair was utilized to train a CNN;multiple networks were concatenated by FC, employed PCA Classifier: SVM
99.5 ± 0.1(URWL)[287]	Descriptor: two deep (10+ layers) networks with supervisory signals, stacked convolution and inception layers. Features extracted from the two networks are concatenated (approximately 30000 dimensions), PCAed to 300 dimensions. Classifier: JB

TABLE 6. Verification performance on LFW.

99.5 ± 0.1(URWL)[287]	Descriptor: two deep (10+ layers) networks with supervisory signals, stacked convolution and inception layers. Features extracted from the two networks are concatenated (approximately 30000 dimensions), PCAed to 300 dimensions. Classifier: JB
88.1 ± 0.6(URLF)[275]	Preprocessing: cropped to 3 different scales and rescale to standard size Descriptor: weighted combination of HT-L2, HT-L3 and V1-like features with model selection Classifier: SVM
92.6 ± 1.4(URLF)[180]	Preprocessing: cropped to 110×150 region, applied Gaussian smooth filter Descriptor: covariance matrix descriptors and soft local binary pattern histograms extracted from regions of varying sizes; WPCAed and concatenated ($13260 \times 2 = 26520$ dimensions) Classifier: two-step metric learning
89.4(IRLF)[144]	Preprocessing: cropped and resized to 110×60, divided in to blocks Descriptor: non-negative sparse codes, sum pooled to construct representation for each block; WPCA for dimension reduction Classifier: metric learning and fusing 8 distances from two scales and four spatial partitions
93.0 ± 0.8(U)[32]	Preprocessing: cropped and resized to 130×90, divide in to 5×2 blocks Descriptor:4-scale LPQ, 5-scale BSIF, processed by EDA+WCCN Classifier: cosine similarity+SVM score fusion

over multiple layers. Moreover, Zhu *et al.* [290], [291] generalized [289] to multi-view perceptron (MVP) to learn 3D face models from 2D images, where identity and view information were encoded by different sets of deterministic and random neurons. These features were combined with pose selective neurons to generate reconstruction feature, which synthesized faces under unobserved viewpoints. This could be beneficial when two persons looked similar in frontal view, but could be better distinguished in other views. MVP could learn rich view representation and better identity features compared with [289].

Supervised hybrid CNN-RBM network for image pairs was built in [264]. It jointly extracted local relational visual features from two face images with hybrid ConvNet RBM. The lower part consisted of 12 groups of 5-layered CNNs, each covered a particular part of face. Then, the outputs were averaged in two layers. First pooling resulted in 5×12 neurons by averaging eight predictions of CNN, and second pooling resulted in 12 neurons by averaging the 5 neurons in the first layer associated with the same group. This hierarchical pooling greatly reduced prediction variance. Finally, a classification RBM was trained with gradient descent to validate whether the pair is of the same class or not. CNNs and RBM were first trained *separately*, and then the whole network was *jointly* fine tuned by backpropagating errors from the top (RBM) to bottom layers (CNNs).

As DL networks extract useful data-adaptive representation without human intervention, proper training strategies should be applied to prevent *overfitting*. Taigman *et al.* [269] used common practices to avoid overfitting: ReLU as activation functions and dropout on first FC. The network was trained on a large Social Face Classification (SFC) dataset with 4.4 million labeled faces, and no overfitting was observed. Huang *et al.* [260] applied *self-taught learning* to utilize a large amount of unlabeled Kyoto natural image dataset and learned representations for the new task via transfer learning. Sun *et al.* [264] selected the model that provided lowest validation error on a separate validation dataset to avoid overfitting. Zhu *et al.* [289] used the deep model to fully reconstruct face in the canonical view, arguing the strong regularization was effective to avoid overfitting.

We categorize DL framework and summarize their features in Table 5.

Deep-ID [266] gave a typical CNN structure for face classification. It operated directly on image pixels and had 4 convolutional layers. Number of filters in the four layers were 20, 40, 60 and 80 respectively. First three layers were max-pooled. The final feature DeepID was of 160 dimensions and could be used to classify via softmax function. A peculiarity of the network is the direct connection of DeepID with the output of the third layer and fourth layer. Deep feature learning based on CNN shared similar framework, though variations such as including supervisory signals [265], [282] could be seen.

VIII. PERFORMANCE ON LFW AND FERET

We selectively list reported performances of proposed recognition algorithms for face identification and verification tasks. For a fair comparison, we make efforts in integrating details of algorithms including preprocessing techniques, feature dimension, dimensional reduction methods and classifying algorithms into our tables.

A. EVALUATION ON LFW

Labeled Face in the Wild (LFW) is a set that contains 13233 training images of 5749 people and is considered as a standard benchmark for face verification. LFW is rather an unconstrained database, in which the only constraint on images is that faces could be detected by the Voila-Jones face detector. Thus LFW contains significant variations in pose, illumination, expression, and occlusion. The evaluation procedure is to divide predefined image pairs into 10 folds of 300 face pairs and for each fold verify

TABLE 7. Verification performance on FERET.

Accuracies(%) on fb/fc/dup1/dup2	Details
97.5/99.5/79.5/77.8[21]	Preprocessing: normalized to 128×128, filter to get 90 Gabor images, divide to 8×8-sized blocks Descriptor: HGPP ($90 \times 8 \times 8 \times 128 = 737280$uint), with FSC to weigh different blocks Classifier: nearest neighbor (rank-1)
99/93/76/78[38]	Preprocessing: cropped to 142×120 pixels Descriptor: constructed by filtering learning followed by fusing similarity scores multi-scale Extended LBP (3bits) with radius 3,5,7 Classifier: Histogram intersection
93.0(train on fa, test on fb)[293]	Preprocessing: cropped to 60×60 pixels Descriptor: multi-manifold analysis Classifier: nearest neighbor (rank-1)
98.4/99.0/82.0/81.6 (E-GV-LBP) 98.4/88.0/82.0/81.6 (GV-LBP-TOP)[108]	Preprocessing: 40 Gabor magnitude, cropped to 88×80 pixels, then partitioned to 11×10 weighted non-overlapping blocks Descriptor: Efficient Gabor Volume LBP (E-GV-LBP) with 8 uniform pattern for each Gabor block ($11 \times 10 \times 40 \times 8 = 35200$int), additionally, GV-LBP-TOP project features to Three Orthogonal Planes (TOP)($11 \times 10 \times 40 \times 8 \times 3 = 105600$int) Classifier: nearest neighbor (rank-1)
99/99/86/85[100]	Preprocessing: cropped to 7×7 blocks, gone through illumination normalization Descriptor: histogram of 512 bins for each block $7 \times 7 \times 512 = 25088$uint
$\approx 97.3/\approx 99.1/\approx 68.4/\approx 67.5$[85]	Preprocessing: cropped to 88×88, filter with 7×4 Gabor filters, divided to 8×8 11×11-sized blocks Descriptor: third order LDP, 4 derivative images of different orientations for each block; 64 bins histogram for quantization ($8 \times 8 \times 7 \times 4 \times 4 \times 64 = 458752$uint) Classifier:nearest neighbor (rank-1)
97/79/66/64[160]	CSU Face Identification Evaluation System with weighted (trained supervisedly) 7×7 block LBP ($7 \times 7 \times 59 = 2891$uint); two training set
91.5/95.4/75.9/72.2[23]	CSU Face Identification Evaluation System with S2FF descriptor (15772 float); two training set split according to [160]
97.2/98.5/85.3/85.5[212]	Preprocessing: cropped and rescaled to 110×110 Descriptor: block-wise HOG, Multi-scale LBP, Gabor (74724 uint) Classifier: extended PLS with rank-1 classifier
98/97/74/71[103]	Preprocessing: normalized to 80×88, divided to 4×8-sized blocks Descriptor: (block) weighted local Gabor binary pattern histogram sequence ($20 \times 11 \times 16 \times 40 = 140800$uint) Classifier: nearest neighbor (rank-1)
97/70/66/50[128]	Preprocessing: divided into 72 blocks Descriptor: used 256 LVP patches of size 3×3 for each block, resulting a representation of dimension $72 \times 256 = 18432$unit (excluding the storage for LVP dictionary) Classifier: nearest neighbor (rank-1)
97/90/71/67[22]	Preprocessing: cropped to 14×12 10×10-sized blocks, filtered the image with 40 Gabor filters Descriptor: built Local Gabor texton dictionary of size 64, thus the descriptor is of dimension $14 \times 12 \times 64 = 10752$uint Classifier: nearest neighbor (rank-1)
98.1/99.0/79.6/79.1[117]	Preprocessing: normalized to 110×110, divided to 10×10 non-overlapping blocks; retina filtering Descriptor: unsigned POEM representation with 3 bins (7 discretized orientations each) and 59 uniform patterns ($10 \times 10 \times 3 \times 59 = 17700$unit) Classifier: nearest neighbor (rank-1)
99.4/100/91.7/90.2[119]	Descriptors: [117] score fused with POD with 64-bin histograms on each of 64 blocks ($17700 + 64 \times 64 = 21796$uint) Classifier: nearest neighbor (rank-1)
99.6/99.5/88.8/85.0[208]	Preprocessing: normalized to 96×96, divided to 8×8 non-overlapping blocks; retina filtering Descriptor: unsigned POEM representation with 3 bins (6 discretized orientations each) and 32 uniform pattern ($8 \times 8 \times 3 \times (32 + 1) = 6336$unit) Classifier: nearest neighbor (rank-1)
99.9/100/93.2/91.0(G-LQP) 99.8/94.3/85.5/78.6(I-LQP)[107]	Preprocessing: equalized; cropped to 90×150, divided to 10×10-sized blocks; use both original and flipped image Descriptor: G-LQP is a compound of 2 Gabor LQP, each with 150 words codebooks ($150 \times 2 \times 9 \times 15 \times 2 = 81000$uint); I-LQP skip the Gabor step ($150 \times 9 \times 15 \times 2 = 40500$uint); WPCA to dimension 900 Classifier: nearest neighbor with cosine measure
99.4/100/91.8/92.3[36]	Preprocessing:divided to 7×7 non-overlapping regions Descriptor: DFD of scale 5 with 1024 dominant patterns ($1024 \times 49 = 50176$float), WPCA reduced to 1100 dimension Classifier: nearest neighbor (rank-1)
99.8/100/93.5/93.2[34]	Preprocessing: aligned and cropped to 128×128 Descriptor: Compact Binary Face Descriptor with local region 8×8 and codebook size 500 (32000int); WPCA to 1000 dimension Classifier: nearest neighbor (rank-1)
99.9/100/95.2/92.7[35]	Preprocessing: cropped to 128×128, divided to 8×8 blocks Descriptor: PDV extracted with radius equals 4, SLBFLE with dictionary size set to 600 ($8 \times 8 \times 600 = 38400$int), WPCA to 500 dimension Classifier: cosine similarity
99.6/99/92/89[123]	Preprocessing: divided to 11×4 non-overlapping regions Descriptor: each region further divided to 5 blocks, 40 Gabor filtered image for each block, each spatial histogram is fix at 8 ($44 \times 5 \times 40 \times 8 = 70400$int) Classifier: ensemble of piecewise Fisher Discriminant Analysis and nearest neighbor with cosine similarity
$\approx 97.1(fb)/\approx 56(dup1)$[124]	Preprocessing: cropped to 64×64, histogram normalization Descriptor: Gabor filtered with 3 scales and 4 orientations, with both magnitude and phase features; estimated histograms in 8×8-sized sub-windows ($24 \times 64 = 1536$int) Classifier: selected 50 features; constructed constrained domain-partitioning RankBoost classifier with rank 1
99/100/84/80[155]	Preprocessing: cropped and illumination normalization; divided to 7×7 blocks Descriptor: DT-LBP with maximum tree depth 13, or 8192bins($8192 \times 49 = 401408$int) Classifier: nearest neighbor with weighted ell_1 distance between histograms
95.4/84.0/74.6/69.2[10]	Descriptor: dense multi-scale HOG; decision-level combination of HOG features extracted from patch sizes 8×8 to 28×28; reduce dimension with LDA Classifier: nearest neighbor with cosine measure
95.5/81.9/60.1/55.6[78]	Preprocessing: cropped to 120×60 Descriptor and classifier: CSU Face Identification Evaluation System with HOG of window size 20×20 on landmarks
99/99/80/78[16]	Preprocessing: cropped to 80×88, divide to 10×8 blocks Descriptor: learned 70 codebooks with sampling step 8×8; code pattern size 5×5, block weighted according to FSC and rareness ($10 \times 8 \times 70 \times 5 \times 5 = 140000$float); dimension reduction with PCA+LDA Classifier: nearest neighbor (rank-1)
99/99/94/93(score-level fusion) 99/99/93/92(feature-level fusion)[111]	Preprocessing: filtered with 40 Gabor filters, divided into 8×8 blocks Descriptors: LGXP with 4 neighbors (feature dimension $40 \times 8 \times 8 \times 2^4 = 40960$uint) and LGBP ($40 \times 8 \times 8 \times 256 = 655360$uint) fusion Classifier: nearest neighbor (rank-1)
99.9/100/95.7/93.1[164]	Preprocessing: cropped to 128×160, divide into 16×16 blocks Descriptors: obtained 720 binary GOM images, aggregated over 8 orientations, formed histogram of 256 bins, compressed histograms with 64 bins uniform part ($16 \times 16 \times 90 \times 64 = 1474560$uint) Classifier: nearest neighbor (rank-1)
99.3/99.5/88.9/84.2 (PCANet-1 trained on Multi-PIE) 99.5/99.0/89.9/86.8 (PCANet-1) 99.7/99.5/88.9/84.2 (PCANet-2 trained on Multi-PIE) 99.6/100/95.4/94.0 (PCANet-2)[169]	Preprocessing: cropped to 150×90 Descriptors: PCANet-1: 1 layer PCANet with 8 5×5-sized filters, aggregated on 15×15-sized non-overlapping blocks(altogether 10×6 blocks), sparse vector with feature dimension$10 \times 6 \times 2^8 = 15360$), WPCAed to 1000 dimensions, PCANet-2 was its 2 layered version, sparse vector with feature dimension$15360 \times 8 = 122880$ Classifier: nearest neighbor (rank-1)
98.1/98.5/81.6/83.2[293]	Preprocessing: cropped to 60×60, divided into 36 10×10-sized blocks Descriptors: built manifold for blocks ($150 \times 130 = 19500$uint), maximized the manifold margins of different persons Classifier: nearest neighbor (rank-1)
98.6/71.1/72.2/47.4[60]	Preprocessing: extracted face with groundtruth eye positions and scaled to 142×120, partitioned to 11×11 non-overlapping blocks Descriptors: extracted 10-scale 8-neighbors uniform LBP ($11 \times 11 \times 10 \times 59 = 71390$unit), used LDA transformation Classifier: nearest neighbor (rank-1)

whether the image pair is of the same person. Table 6 lists accuracies of LFW under different protocols. We enumerate protocols used in LFW evaluation procedure as follows: Unsupervised (U), Image-Restricted, No Outside Data (IRNO), Unrestricted, No Outside Data (URNO), Image-Restricted, Label-Free Outside Data (IRLF), Unrestricted, Label-Free Outside Data (URLF), Unrestricted With Labeled Outside Data (URWL). Details of protocols can be seen in [292].It should be noted that [36] does not follow protocols above, in that it uses FERET database for training, but training is done in an unsupervised way (Unsupervised with Label-Free Outside data, ULF).

We note some cases that utilize outside data for LFW verification task. Yi *et al.* [17] utilized outside data for alignment. Detection and alignment of [218] were based on images collected from Yahoo News. In order to learn sparse projection matrix to map extremely high-dimensional feature to low-dimensional ones, [29], [213], [214] trained on WDRef and tested on LFW evaluating under URWL protocol. Deep architectures are prone to overfitting so larger outside databases are often used in the training phase. Taigman *et al.* [269] trained on Social Face Classification (SFC) dataset and tested on LFW. Zhu *et al.* [277] trained on CelebFaces dataset. Sun *et al.* [266] trained DeepID on CelebFace+, while Sun *et al.* [265] trained DeepID2 on two complementary subset of CelebFace+. Sun *et al.* [282], [287] enlarged training data by merging CelebFaces+ with WDRef dataset. Zhou *et al.* [276] collected and label many celebrities from Internet to build Megvii Face Classification database that has 5 million labeled faces with approximate 20000 individuals.

We give trivia that does not integrate into Table 6. Sharma *et al.* [37] aligned the images with manually labeled images that do not intersect with LFW, then they learned GMM on randomly sampled 3×3 pixel differential vectors and computed higher-order Fisher score. Lei *et al.* [108] applied Efficient Gabor Volume LBP on 40 Gabor magnitude images; trails of different statistical uniform patterns were done and the number was finally set to 8. Sun *et al.* [266] compared results of CNN with Joint Bayesian Classifier (JB) and Neural Networks Classifier (NN) and found out JB outperforms NN. Sun *et al.* [265], [266] trained from a particular patch and its horizontally flipped counterpart, thus forming *two* DeepID/DeepID2 vectors.

B. EVALUATION ON FERET

The FERET database is an SSPP gallery composed of 14051 facial images. The set contains a gallery set of 1196 individuals, and four testing set with variations on lighting, facial expressions, pose, and age. Accuracies and details of face identification procedure are given in Table 7.

IX. CONCLUSION

In this paper, we reviewed over 200 works regarding feature extraction for facial images. Extraction of proper features are the core bridging the gap of local features and discriminative representation of faces. We categorize these works into: filtering and local features, feature encoding, spatial pooling and holistic feature processing. We see deep learning as an automatic and multi-layer version of the shallow ones. By analyzing the characteristics of facial structures we further summarize motives and common practices in preprocessing, integrating local feature position and scales, fusing multiple features, discriminant feature extraction and other alternations to the framework and its building blocks. Finally, we give examples of the closely linked systems that output robust feature representation and show their experiment results and settings.

ACKNOWLEDGMENT

The authors were with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications, Beijing, China.

REFERENCES

[1] L. Best-Rowden, H. Han, C. Otto, B. F. Klare, and A. K. Jain, "Unconstrained face recognition: Identifying a person of interest from a media collection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2144–2157, Dec. 2014.

[2] Y. Huang, Z. Wu, L. Wang, and T. Tan, "Feature coding in image classification: A comprehensive study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 493–506, Mar. 2014.

[3] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 10, pp. 1042–1052, Oct. 1993.

[4] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Amer. A, Opt. Image Sci. Vis.*, vol. 4, no. 3, pp. 519–524, 1987.

[5] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[6] J. Zou, Q. Ji, and G. Nagy, "A comparative study of local matching approach for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2617–2628, Oct. 2007.

[7] B. Yang and S. Chen, "A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image," *Neurocomputing*, vol. 120, no. 23, pp. 365–379, Nov. 2013.

[8] K.-C. Song, Y.-H. Yan, W.-H. Chen, and X. Zhang, "Research and perspective on local binary pattern," *Acta Autom. Sinica*, vol. 39, no. 6, pp. 730–744, 2013.

[9] A. Suruliandi, K. Meena, and R. R. Rose, "Local binary pattern and its derivatives for face recognition," *IET Comput. Vis.*, vol. 6, no. 5, pp. 480–488, 2012.

[10] O. Déniz, G. Bueno, J. Salido, and F. de la Torre, "Face recognition using histograms of oriented gradients," *Pattern Recognit. Lett.*, vol. 32, no. 12, pp. 1598–1603, 2011.

[11] T. Ahonen and M. Pietikäinen, "Image description using joint distribution of filter bank responses," *Pattern Recognit. Lett.*, vol. 30, no. 4, pp. 368–376, 2009.

[12] C. Ding and D. Tao. (Feb. 2015). "A comprehensive survey on pose-invariant face recognition." [Online]. Available: https://arxiv.org/abs/1502.04383

[13] L. Wiskott, J.-M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 775–779, Jul. 1997.

[14] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[15] V. Ghosal, P. Tikmani, and P. Gupta, "Face classification using Gabor wavelets and random forest," in *Proc. Can. Conf. Comput. Robot Vis. (CRV)*, 2009, pp. 68–73.

[16] S. Xie, S. Shan, X. Chen, X. Meng, and W. Gao, "Learned local Gabor patterns for face representation and recognition," *Signal Process.*, vol. 89, no. 12, pp. 2333–2344, 2009.

[17] D. Yi, Z. Lei, and S. Z. Li, "Towards pose robust face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3539–3545.
[18] M. Yang and L. Zhang, "Gabor feature based sparse representation for face recognition with Gabor occlusion dictionary," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2010, pp. 448–461.
[19] A. Li, S. Shan, X. Chen, and W. Gao, "Cross-pose face recognition based on partial least squares," *Pattern Recognit. Lett.*, vol. 32, no. 15, pp. 1948–1955, 2011.
[20] C. Wang, Z. Chai, and Z. Sun, "Face recognition using histogram of co-occurrence Gabor phase patterns," in *Proc. 20th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 2777–2781.
[21] B. Zhang, S. Shan, X. Chen, and W. Gao, "Histogram of Gabor phase patterns (HGPP): A novel object representation approach for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 57–68, Jan. 2007.
[22] Z. Lei, S. Z. Li, R. Chu, and X. Zhu, "Face recognition with local Gabor textons," in *Advances in Biometrics*. New York, NY, USA: Springer, 2007, pp. 49–57.
[23] E. Meyers and L. Wolf, "Using biologically inspired features for face processing," *Int. J. Comput. Vis.*, vol. 76, no. 1, pp. 93–104, 2008.
[24] N. Pinto, J. J. DiCarlo, and D. D. Cox, "How far can you get with a modern face recognition test set using only simple features?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 2591–2598.
[25] M. Heikkilä and M. Pietikäinen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.
[26] K. Jeong, J. Choi, and G. J. Jang, "Semi-local structure patterns for robust face detection," *IEEE Signal Process. Lett.*, vol. 22, no. 9, pp. 1400–1403, Sep. 2015.
[27] T. Chakraborti and A. Chatterjee, "A novel binary adaptive weight GSA based feature selection for face recognition using local gradient patterns, modified census transform, and local binary patterns," *Eng. Appl. Artif. Intell.*, vol. 33, pp. 80–90, Aug. 2014.
[28] C. Lu, D. Zhao, and X. Tang, "Face recognition using face patch networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3288–3295.
[29] D. Chen, X. Cao, F. Wen, and J. Sun, "Blessing of dimensionality: High-dimensional feature and its efficient compression for face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3025–3032.
[30] X. Zhu, Z. Lei, J. Yan, D. Yi, and S. Z. Li, "High-fidelity pose and expression normalization for face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 787–796.
[31] M. Guillaumin, J. Verbeek, and C. Schmid, "Is that you? Metric learning approaches for face identification," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 498–505.
[32] A. Ouamane, B. Messaoud, A. Guessoum, A. Hadid, and M. Cheriet, "Multi scale multi descriptor local binary features and exponential discriminant analysis for robust face authentication," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 313–317.
[33] S. R. Arashloo and J. Kittler, "Class-specific kernel fusion of multiple descriptors for face verification using multiscale binarised statistical image features," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2100–2109, Dec. 2014.
[34] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2041–2056, Oct. 2015.
[35] J. Lu, V. Erin Liong, and J. Zhou, "Simultaneous local binary feature learning and encoding for face recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3721–3729.
[36] Z. Lei, M. Pietikäinen, and S. Z. Li, "Learning discriminant face descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 289–302, Feb. 2014.
[37] G. Sharma, S. ul Hussain, and F. Jurie, "Local higher-order statistics (LHS) for texture categorization and facial analysis," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 1–12.
[38] Z. Lei, D. Yi, and S. Z. Li, "Discriminant image filter learning for face recognition with local binary pattern like representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2512–2517.
[39] Z. Guo and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE Trans. Image Process.*, vol. 19, no. 6, pp. 1657–1663, Jan. 2010.
[40] C.-K. Tran, T.-F. Lee, C.-C. Tuan, C.-H. Lu, and P.-J. Chao, "Improving face recognition performance using similarity feature-based selection and classification algorithm," in *Proc. 2nd Int. Conf. Robot, Vis. Signal Process. (RVSP)*, Dec. 2013, pp. 56–60.
[41] B. Jun and D. Kim, "Robust face detection using local gradient patterns and evidence accumulation," *Pattern Recognit.*, vol. 45, no. 9, pp. 3304–3316, 2012.
[42] S. Liao, Z. Lei, S. Z. Li, X. Yuan, and R. He, "Structured ordinal features for appearance-based object representation," in *Analysis and Modeling of Faces and Gestures*. New York, NY, USA: Springer, 2007, pp. 183–192.
[43] Z. Lei, D. Yi, and S. Z. Li, "Local gradient order pattern for face representation and recognition," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 387–392.
[44] W. Zhang, X. Wang, and X. Tang, "Coupled information-theoretic encoding for face photo-sketch recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 513–520.
[45] Z. Cao, Q. Yin, X. Tang, and J. Sun, "Face recognition with learning-based descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2707–2714.
[46] J. Gu and C. Liu, "Feature local binary patterns with application to eye detection," *Neurocomputing*, vol. 113, pp. 138–152, Aug. 2013.
[47] Z. Li, G. Liu, Y. Yang, and J. You, "Scale- and rotation-invariant local binary pattern using scale-adaptive texton and subuniform-based circular shift," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2130–2140, Apr. 2012.
[48] S. ul Hussain and B. Triggs, "Visual recognition using local quantized patterns," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 716–729.
[49] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and Modeling of Faces and Gestures*. New York, NY, USA: Springer, 2007, pp. 168–182.
[50] M. A. Akhloufi and A. Bendada, "Locally adaptive texture features for multispectral face recognition," in *Proc. IEEE Int. Conf. Syst., Man Cybern. (SMC)*, Oct. 2010, pp. 3308–3314.
[51] J. Zhang, J. Liang, and H. Zhao, "Local energy pattern for texture classification using self-adaptive quantization thresholds," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 31–42, Jan. 2013.
[52] J. Ren, X. Jiang, and J. Yuan, "Noise-resistant local binary pattern with an embedded error-correction mechanism," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4049–4060, Oct. 2013.
[53] T. Ahonen and M. Pietikäainen, "Soft histograms for local binary patterns," in *Proc. Finnish signal Process. Symp. (FINSIG)*, vol. 5. 2007, pp. 1–4.
[54] H. Ye, R. Hu, H. Yu, and R. I. Damper, "Face recognition based on adaptive soft histogram local binary patterns," in *Biometric Recognition*. New York, NY, USA: Springer, 2013, pp. 62–70.
[55] C. Ding, J. Choi, D. Tao, and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 518–531, Mar. 2016.
[56] C. Ding, C. Xu, and D. Tao, "Multi-task pose-invariant face recognition," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 980–993, Mar. 2015.
[57] V. Ojansivu and J. Heikkilä, "Blur insensitive texture classification using local phase quantization," in *Proc. Int. Conf. Image Signal Process.*, 2008, pp. 236–243.
[58] E. Sariyanidi, H. Gunes, and A. Cavallaro, "Automatic analysis of facial affect: A survey of registration, representation, and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1113–1133, Jun. 2015.
[59] C. H. Chan, J. Kittler, N. Poh, T. Ahonen, and M. Pietikainen, "(Multiscale) local phase quantisation histogram discriminant analysis with score normalisation for robust face recognition," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops (ICCV Workshops)*, Sep. 2009, pp. 633–640.
[60] C.-H. Chan, J. Kittler, and K. Messer, "Multi-scale local binary pattern histograms for face recognition," in *Proc. Int. Conf. Biometrics (ICB)*, 2007, pp. 809–818.
[61] M. Dahmane and L. Gagnon, "Local phase-context for face recognition under varying conditions," *Proc. Comput. Sci.*, vol. 39, pp. 12–19, Jan. 2014.
[62] B. Froba and A. Ernst, "Face detection with the modified census transform," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, May 2004, pp. 91–96.

[63] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2. Sep. 1999, pp. 1150–1157.
[64] A. Bosch, A. Zisserman, and X. Muñoz, "Scene classification via pLSA," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2006, pp. 517–530.
[65] J. Hu, J. Lu, and Y.-P. Tan, "Discriminative deep metric learning for face verification in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1875–1882.
[66] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1794–1801.
[67] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Kernel sparse representation for image classification and face recognition," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2010, pp. 1–14.
[68] K. Simonyan, O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Fisher vector faces in the wild," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–13.
[69] A. Vinay *et al.*, "Face recognition using VLAD and its variants," in *Proc. 6th Int. Conf. Comput. Commun. Technol.*, 2015, pp. 233–238.
[70] O. M. Parkhi, K. Simonyan, A. Vedaldi, and A. Zisserman, "A compact and discriminative face track descriptor," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1693–1700.
[71] H. Li, G. Hua, Z. Lin, J. Brandt, and J. Yang, "Probabilistic elastic matching for pose variant face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3499–3506.
[72] H. Li, G. Hua, X. Shen, Z. Lin, and J. Brandt, "Eigen-PEP for video face recognition," in *Computer Vision—ACCV*. New York, NY, USA: Springer, 2015, pp. 17–33.
[73] H. Li and G. Hua, "Hierarchical-PEP model for real-world face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4055–4064.
[74] C. Baecchi, F. Turchini, L. Seidenari, A. D. Bagdanov, and A. D. Bimbo, "Fisher vectors over random density forests for object recognition," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 4328–4333.
[75] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 404–417.
[76] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2005, pp. 886–893.
[77] M. M. Abdelwahab, I. Yousry, and W. Mikhael, "A novel algorithm for simultaneous face detection and recognition," in *Proc. IEEE 55th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2012, pp. 670–673.
[78] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using HOG–EBGM," *Pattern Recognit. Lett.*, vol. 29, no. 10, pp. 1537–1543, 2008.
[79] T. Berg and P. N. Belhumeur, "POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 955–962.
[80] X. Zhang, M. H. Mahoor, and S. M. Mavadati, "Facial expression recognition using l_p-norm MKL multiclass-SVM," *Mach. Vis. Appl.*, vol. 26, no. 4, pp. 467–483, 2015.
[81] W. Hwang, G. Park, J. Lee, and S.-C. Kee, "Multiple face model of hybrid Fourier feature for large face image set," in *Proc. IEEE Comput. Society Conf. Comput. Vis. Pattern Recognit.*, vol. 2. Jun. 2006, pp. 1574–1581.
[82] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Advances in Biometrics*. New York, NY, USA: Springer, 2009, pp. 199–208.
[83] Y. Wong, M. T. Harandi, and C. Sanderson, "On robust face recognition via sparse coding: The good, the bad and the ugly," *IET Biometrics*, vol. 3, no. 4, pp. 176–189, 2014.
[84] M. Uzun-Per and M. Gokmen, "Face recognition with a novel image representation: Local walsh-Hadamard transform," in *Proc. 5th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Dec. 2014, pp. 1–6.
[85] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with high-order local pattern descriptor," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 533–544, Feb. 2010.
[86] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local tetra patterns: A new feature descriptor for content-based image retrieval," *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2874–2886, May 2012.
[87] G. Tao, X. L. Feng, F. Chen, and J. H. Zhai, "Local comprehensive patterns: A novel face feature descriptor," *Optik-Int. J. Light Electron Opt.*, vol. 124, no. 24, pp. 7022–7026, 2013.
[88] Z. Xie and Z. Wang, "Joint encoding of multi-scale LBP for infrared face recognition," in *Genetic and Evolutionary Computing*. Springer, 2015, pp. 269–276.
[89] T.-T. Do and E. Kijak, "Face recognition using co-occurrence histograms of oriented gradients," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1301–1304.
[90] H. J. Seo and P. Milanfar, "Face verification using the LARK representation," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 4, pp. 1275–1286, Dec. 2011.
[91] A. Yao and S. Yu, "Robust face representation using hybrid spatial feature interdependence matrix," *IEEE Trans. Image Process.*, vol. 22, no. 8, pp. 3247–3259, Aug. 2013.
[92] J. Ren, X. Jiang, and J. Yuan, "Learning LBP structure by maximizing the conditional mutual information," *Pattern Recognit.*, vol. 48, no. 10, pp. 3180–3190, 2015.
[93] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1. Dec. 2001, pp. I-511–I-518.
[94] H. Han, S. Shan, X. Chen, and W. Gao, "A comparative study on illumination preprocessing in face recognition," *Pattern Recognit.*, vol. 46, no. 6, pp. 1691–1699, Jun. 2013.
[95] S. Nigam and A. Khare, "Multiscale local binary patterns for facial expression-based human emotion recognition," in *Computational Vision and Robotics*. New York, NY, USA: Springer, 2015, pp. 71–77.
[96] Z. Liu and C. Liu, "Robust face recognition using color information," in *Advances in Biometrics*. New York, NY, USA: Springer, 2009, pp. 122–131.
[97] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) for face recognition," in *Int. Conf. Consum. Electron. (ICCE) Dig. Tech. Papers*, Jan. 2010, pp. 329–330.
[98] F. Zhong and J. Zhang, "Face recognition with enhanced local directional patterns," *J. Neurocomput.*, vol. 119, pp. 375–384, Nov. 2013.
[99] M. J. Jones and P. Viola, "Face recognition using boosted local features," Mitsubishi Electr. Res. Lab., Cambridge, MA, USA, Tech. Rep. TR2003-25, 2003.
[100] D. Maturana, D. Mery, and A. Soto, "Learning discriminative local binary patterns for face recognition," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops (FG)*, Mar. 2011, pp. 470–475.
[101] R. Kumar, A. Banerjee, and B. C. Vemuri, "Volterrafaces: Discriminant analysis using volterra kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 150–155.
[102] J. Kannala and E. Rahtu, "BSIF: Binarized statistical image features," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 1363–1366.
[103] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Oct. 2005, pp. 786–791.
[104] W. Zhang, S. Shan, H. Zhang, W. Gao, and X. Chen, "Multi-resolution histograms of local variation patterns (MHLVP) for robust face recognition," in *Audio- and Video-Based Biometric Person Authentication*. New York, NY, USA: Springer, 2005, pp. 937–944.
[105] C. Zhang, G. Yuzhang, Z. Zhang, and Z. Yunlong, "Multi-orientation log-Gabor local binary pattern for face representation and recognition," *IEICE Trans. Inf. Syst.*, vol. E98-D, no. 2, pp. 448–452, 2015.
[106] S.-R. Zhou, J.-P. Yin, and J.-M. Zhang, "Local binary pattern (LBP) and local phase quantization (LBQ) based on Gabor filter for face representation," *Neurocomputing*, vol. 116, pp. 260–264, Sep. 2013.
[107] S. U. Hussain, T. Napoléon, and F. Jurie, "Face recognition using local quantized patterns," in *Proc. Brit. Machive Vis. Conf.*, 2012, p. 11.
[108] Z. Lei, S. Liao, M. Pietikäinen, and S. Z. Li, "Face recognition by exploring information jointly in space, scale and orientation," *IEEE Trans. Image Process.*, vol. 20, no. 1, pp. 247–256, Jan. 2011.
[109] W. Zhang, S. Shan, L. Qing, X. Chen, and W. Gao, "Are Gabor phases really useless for face recognition?" *Pattern Anal. Appl.*, vol. 12, no. 3, pp. 301–307, 2009.
[110] Y. Guo and L. Zhang, "Face recognition algorithm based on uniform LGBP and SRC," *J. Comput.-Aided Des. Comput. Graph.*, vol. 27, no. 3, pp. 400–405, 2014.
[111] S. Xie, S. Shan, X. Chen, and J. Chen, "Fusing local patterns of Gabor magnitude and phase for face recognition," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1349–1361, May 2010.

[112] J. Y. Choi, K. N. Plataniotis, and Y. M. Ro, "Using colour local binary pattern features for face recognition," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 4541–4544.
[113] S. H. Lee, J. Y. Choi, Y. M. Ro, and K. N. Plataniotis, "Local color vector binary patterns from multichannel face images for face recognition," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2347–2353, Apr. 2012.
[114] V. Jain, J. L. Crowley, and A. Lux, "Local binary patterns calculated over Gaussian derivative images," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 3987–3992.
[115] L. Zhou, W. Liu, Z.-M. Lu, and T. Nie, "Face recognition based on curvelets and local binary pattern features via using local property preservation," *J. Syst. Softw.*, vol. 95, pp. 209–216, Sep. 2014.
[116] S. Zhao, Y. Gao, and B. Zhang, "Sobel-LBP," in *Proc. 15th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2008, pp. 2144–2147.
[117] N.-S. Vu and A. Caplier, "Face recognition with patterns of oriented edge magnitudes," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2010, pp. 313–326.
[118] N.-S. Vu and A. Caplier, "Mining patterns of orientations and magnitudes for face recognition," in *Proc. Int. Joint Conf. Biometrics (IJCB)*, 2011, pp. 1–8.
[119] N.-S. Vu, "Exploring patterns of gradient orientations and magnitudes for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 2, pp. 295–304, Feb. 2013.
[120] F. Juefei-Xu and M. Savvides, "Subspace-based discrete transform encoded local binary patterns representations for robust periocular matching on NIST's face recognition grand challenge," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3490–3505, Aug. 2014.
[121] J. Liu, X. Jing, S. Sun, and Z. Lian, "Variable length dominant Gabor local binary pattern (VLD-GLBP) for face recognition," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 89–92.
[122] K.-J. Jeong and D.-J. Kim, "Face recognition by weighted multi-resolution uniform local Gabor binary patterns," in *Proc. 12th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2012, pp. 2167–2170.
[123] S. Shan, W. Zhang, Y. Su, X. Chen, and W. Gao, "Ensemble of piecewise FDA based on spatial histograms of local (Gabor) binary patterns for face recognition," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 4. Aug. 2006, pp. 606–609.
[124] B. Yao, H. Ai, Y. Ijiri, and S. Lao, "Domain-partitioning rankboost for face recognition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, vol. 1. Sep. 2007, pp. I-129–I-132.
[125] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural Comput.*, vol. 10, no. 2, pp. 251–276, 1998.
[126] Y. Freund, S. Dasgupta, M. Kabra, and N. Verma, "Learning the structure of manifolds using random projections," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 473–480.
[127] O. Nikisins and M. Greitans, "Reduced complexity automatic face recognition algorithm based on local binary patterns," in *Proc. 19th Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Apr. 2012, pp. 433–436.
[128] X. Meng, S. Shan, X. Chen, and W. Gao, "Local visual primitives (LVP) for face modelling and recognition," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2. Aug. 2006, pp. 536–539.
[129] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 3304–3311.
[130] M. Faraki, M. T. Harandi, and F. Porikli, "More about VLAD: A leap from Euclidean to Riemannian manifolds," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4951–4960.
[131] A. Li, V. Morariu, and L. S. Davis, "Selective encoding for recognizing unreliably localized faces," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 3613–3621.
[132] C. Lu and X. Tang. (Apr. 2014). "Surpassing human-level face verification performance on LFW with gaussianface." [Online]. Available: https://arxiv.org/abs/1404.3840
[133] N. D. Lawrence, "Gaussian process latent variable models for visualisation of high dimensional data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, vol. 16. no. 3, pp. 329–336.
[134] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.
[135] F. Shen and C. Shen. (Sep. 2013). "Generic image classification approaches excel on face recognition." [Online]. Available: https://arxiv.org/abs/1309.5594
[136] Z. Cui, S. Shan, X. Chen, and L. Zhang, "Sparsely encoded local descriptor for face recognition," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops (FG)*, Mar. 2011, pp. 149–154.
[137] J.-C. Chen, V. M. Patel, H. T. Ho, and R. Chellappa, "Dictionary-based video face recognition using dense multi-scale facial landmark features," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 733–737.
[138] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, Sep. 2014.
[139] W. Deng, J. Hu, and J. Guo, "Extended SRC: Undersampled face recognition via intraclass variant dictionary," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1864–1870, Sep. 2012.
[140] P. Zhu, M. Yang, L. Zhang, and I.-Y. Lee, "Local generic representation for face recognition with single sample per person," in *Computer Vision—ACCV*. New York, NY, USA: Springer, 2015, pp. 34–50.
[141] S. Gao, K. Jia, L. Zhuang, and Y. Ma, "Neither global nor local: Regularized patch-based representation for single sample per person face recognition," *Int. J. Comput. Vis.*, vol. 111, no. 3, pp. 365–383, 2015.
[142] R. Min and J.-L. Dugelay, "Improved combination of LBP and sparse representation based classification (SRC) for face recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2011, pp. 1–6.
[143] L. Zini, N. Noceti, G. Fusco, and F. Odone, "Structured multi-class feature selection with an application to face recognition," *Pattern Recognit. Lett.*, vol. 55, pp. 35–41, Apr. 2015.
[144] Z. Cui, W. Li, D. Xu, S. Shan, and X. Chen, "Fusing robust face region descriptors via multiple metric learning for face recognition in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 3554–3561.
[145] H. Guo, R. Wang, J. Choi, and L. S. Davis, "Face verification using sparse representations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2012, pp. 37–44.
[146] F. Liu, J. Tang, Y. Song, X. Xiang, and Z. Tang, "Local structure based sparse representation for face recognition with single sample per person," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 713–717.
[147] I. Theodorakopoulos, I. Rigas, G. Economou, and S. Fotopoulos, "Face recognition via local sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 1647–1652.
[148] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 471–478.
[149] Q. Shi, A. Eriksson, A. van den Hengel, and C. Shen, "Is face recognition really a compressive sensing problem?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 553–560.
[150] P. Zhu, L. Zhang, Q. Hu, and S. C. Shiu, "Multi-scale patch based collaborative representation for face recognition with margin distribution optimization," in *Computer Vision—ECCV*. New York, NY, USA: Springer, 2012, pp. 822–835.
[151] R.-X. Ding, H. Huang, and J. Shang, "Patch-based locality-enhanced collaborative representation for face recognition," *IET Image Process.*, vol. 9, no. 3, pp. 211–217, 2015.
[152] B.-C. Chen, Y.-H. Kuo, Y.-Y. Chen, K.-Y. Chu, and W. Hsu, "Semi-supervised face image retrieval using sparse coding with identity constraint," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1369–1372.
[153] E. Nowak and F. Jurie, "Learning visual similarity measures for comparing never seen objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2007, pp. 1–8.
[154] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.
[155] D. Maturana, D. Mery, and Á. Soto, "Face recognition with decision tree-based local binary patterns," in *Computer Vision—ACCV*. New York, NY, USA: Springer, 2011, pp. 618–629.
[156] C. Orrite, A. Gañán, and G. Rogez, "HOG-based decision tree for facial expression classification," in *Pattern Recognition and Image Analysis*. New York, NY, USA: Springer, 2009, pp. 176–183.
[157] D. Gong, Z. Li, D. Tao, J. Liu, and X. Li, "A maximum entropy feature descriptor for age invariant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 5289–5297.
[158] J. Wright and G. Hua, "Implicit elastic matching with random projections for pose-variant face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 1502–1509.
[159] Y. Huang, Z. Wu, L. Wang, and C. Song, "Multiple spatial pooling for visual object recognition," *Neurocomputing*, vol. 129, pp. 225–231, Apr. 2014.

[160] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision—ECCV*. New York, NY, USA: Springer, 2004, pp. 469–481.

[161] A. Chouchane, M. Belahcene, A. Ouamane, and S. Bourennane, "3D face recognition based on histograms of local descriptors," in *Proc. 4th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Oct. 2014, pp. 1–5.

[162] Y. Fang and Z. Wang, "Improving LBP features for gender classification," in *Proc. Int. Conf. Wavelet Anal. Pattern Recognit. (ICWAPR)*, vol. 1. Aug. 2008, pp. 373–377.

[163] J. Qian, J. Yang, and Y. Xu, "Local structure-based image decomposition for feature extraction with applications to face recognition," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3591–3603, Sep. 2013.

[164] Z. Chai, Z. Sun, H. Méndez-Vázquez, R. He, and T. Tan, "Gabor ordinal measures for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 1, pp. 14–26, Jan. 2014.

[165] J. Wang *et al.*, "Locality constrained joint dynamic sparse representation for local matching based face recognition," *PLoS ONE*, vol. 9, no. 11, p. e113198, 2014.

[166] Q. Hu, X. Peng, P. Yang, F. Yang, and D. N. Metaxas, "Robust multi-pose facial expression recognition," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 1782–1787.

[167] C.-K. Tran, T.-F. Lee, L. Chang, and P.-J. Chao, "Face description with local binary patterns and local ternary patterns: Improving face recognition performance using similarity feature-based selection and classification algorithm," in *Proc. Int. Symp. Comput., Consum. Control (IS3C)*, Jun. 2014, pp. 520–524.

[168] S. Biswas and J. Sil, "Facial expression recognition using modified local binary pattern," in *Computational Intelligence in Data Mining*, vol. 2. New York, NY, USA: Springer, 2015, pp. 595–604.

[169] T.-H. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. (Apr. 2014). "PCANet: A simple deep learning baseline for image classification?" [Online]. Available: https://arxiv.org/abs/1404.3606

[170] B. Yuan, H. Cao, and J. Chu, "Combining local binary pattern and local phase quantization for face recognition," in *Proc. Int. Symp. Biometrics Secur. Technol. (ISBAST)*, Mar. 2012, pp. 51–53.

[171] S. Chowdhury, J. K. Sing, D. K. Basu, and M. Nasipuri, "Face recognition by fusing local and global discriminant features," in *Proc. 2nd Int. Conf. Emerg. Appl. Inf. Technol. (EAIT)*, Feb. 2011, pp. 102–105.

[172] B. Jiang, B. Martinez, M. F. Valstar, and M. Pantic, "Decision level fusion of domain specific regions for facial action recognition," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 1776–1781.

[173] H. Tang, B. Yin, Y. Sun, and Y. Hu, "3D face recognition using local binary patterns," *Signal Process.*, vol. 93, no. 8, pp. 2190–2198, Aug. 2013.

[174] I. Masi, C. Ferrari, A. D. Bimbo, and G. Medioni, "Pose independent face recognition by localizing local binary patterns via deformation components," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 4477–4482.

[175] Y. Jiang, B. Wang, Y. Zhou, W. Li, and Q. Liao, "Patterns of weber magnitude and orientation for uncontrolled face representation and recognition," *Neurocomputing*, vol. 165, pp. 190–201, Oct. 2015.

[176] X. Li, Q. Ruan, Y. Jin, G. An, and R. Zhao, "Fully automatic 3D facial expression recognition using polytypic multi-block local binary patterns," *Signal Process.*, vol. 108, pp. 297–308, Mar. 2015.

[177] M. S. Aslan, Z. Hailat, T. K. Alafif, and X.-W. Chen, "Multi-channel multi-model feature learning for face recognition," *Pattern Recognit. Lett.*, vol. 85, pp. 79–83, Jun. 2017.

[178] C. Sanderson, M. T. Harandi, Y. Wong, and B. C. Lovell, "Combined learning of salient local descriptors and distance metrics for image set face verification," in *Proc. IEEE 9th Int. Conf. Adv. Video Signal-Based Surveill. (AVSS)*, Sep. 2012, pp. 294–299.

[179] R. Kumar, A. Banerjee, B. C. Vemuri, and H. Pfister, "Maximizing all margins: Pushing face recognition with Kernel Plurality," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2375–2382.

[180] C. Huang, S. Zhu, and K. Yu. (Dec. 2012). "Large scale strongly supervised ensemble metric learning, with applications to face verification and retrieval." [Online]. Available: https://arxiv.org/abs/1212.6094

[181] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1885–1896, Aug. 2009.

[182] Z. Lei, D. Yi, and S. Z. Li, "Learning stacked image descriptor for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 9, pp. 1685–1696, Sep. 2016.

[183] X. Zhai, Y. Peng, and J. Xiao, "PDSS: Patch-descriptor-similarity space for effective face verification," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 961–964.

[184] X.-M. Ren, X.-F. Wang, and Y. Zhao, "An efficient multi-scale overlapped block LBP approach for leaf image recognition," in *Intelligent Computing Theories and Applications*. New York, NY, USA: Springer, 2012, pp. 237–243.

[185] H. K. Galoogahi and T. Sim, "Face sketch recognition by Local Radon Binary Pattern: LRBP," in *Proc. 19th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2012, pp. 1837–1840.

[186] F. Shen, C. Shen, and H. T. Shen. (Jun. 2014). "Face image classification by pooling raw features." [Online]. Available: https://arxiv.org/abs/1406.6811

[187] T. Mäenpää and M. Pietikäinen, "Multi-scale binary patterns for texture analysis," in *Image Analysis*. New York, NY, USA: Springer, 2003, pp. 885–892.

[188] S. Yan, H. Wang, X. Tang, and T. Huang, "Exploring feature descritors for face recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, vol. 1. Apr. 2007, pp. I-629–I-632.

[189] C. Shan, "Learning local binary patterns for gender classification on real-world face images," *Pattern Recognit. Lett.*, vol. 33, no. 4, pp. 431–437, 2012.

[190] S. R. Arashloo and J. Kittler, "Efficient processing of MRFs for unconstrained-pose face recognition," in *Proc. IEEE 6th Int. Conf. Biometrics, Theory, Appl. Syst. (BTAS)*, Sep. 2013, pp. 1–8.

[191] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz, "Fast high dimensional vector multiplication face recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1960–1967.

[192] S. Nikan and M. Ahmadi, "Local gradient-based illumination invariant face recognition using local phase quantisation and multi-resolution local binary pattern fusion," *IET Image Process.*, vol. 9, no. 1, pp. 12–21, 2015.

[193] S. Liao, X. Zhu, Z. Lei, L. Zhang, and S. Z. Li, "Learning multi-scale block local binary patterns for face recognition," in *Advances in Biometrics*. New York, NY, USA: Springer, 2007, pp. 828–837.

[194] Z. Guo, L. Zhang, D. Zhang, and X. Mou, "Hierarchical multiscale LBP for face and palmprint recognition," in *Proc. 17th IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 4521–4524.

[195] Z. Liu, X. Song, and Z. Tang, "Fusing hierarchical multi-scale local binary patterns and virtual mirror samples to perform face recognition," *Neural Comput. Appl.*, vol. 26, pp. 2013–2026, Nov. 2015.

[196] Z. Lei and S. Z. Li, "Fast multi-scale local phase quantization histogram for face recognition," *Pattern Recognit. Lett.*, vol. 33, no. 13, pp. 1761–1767, 2012.

[197] D. Nan, Z. Xu, and S. Bian, "Face recognition based on multi-classifierweighted optimization and sparse representation," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 6, no. 5, pp. 423–436, 2013.

[198] B. Yang, X.-H. Wang, X. Yang, and X.-X. Huang, "Face recognition method based on HOG pyramid," *J. Zhejiang Univ. Eng. Sci.*, vol. 48, pp. 1564–1569 and 1681, Sep. 2014.

[199] V. Štruc, J. Z. Gros, S. Dobrišek, and N. Pavešic, "Exploiting representation plurality for robust and efficient face recognition," in *Proc. 22nd Int. Electrotech. Comput. Sci. Conf. (ERK)*, 2013, pp. 121–124.

[200] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[201] L. Liu, L. Zhao, Y. Long, G. Kuang, and P. Fieguth, "Extended local binary patterns for texture classification," *Image Vis. Comput.*, vol. 30, no. 2, pp. 86–99, Feb. 2012.

[202] B. Ma, Y. Su, and F. Jurie, "Local descriptors encoded by fisher vectors for person re-identification," in *Computer Vision—ECCV. Workshops and Demonstrations*. Berlin, Germany: Springer, 2012, pp. 413–422.

[203] X. Tan and B. Triggs, "Fusing Gabor and LBP feature sets for kernel-based face recognition," in *Analysis and Modeling of Faces and Gestures*. New York, NY, USA: Springer, 2007, pp. 235–249.

[204] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.

[205] I. M. de Diego, Á. Serrano, C. Conde, and E. Cabello, "Face verification with a kernel fusion method," *Pattern Recognit. Lett.*, vol. 31, no. 9, pp. 837–844, 2010.

[206] J. Yang and J.-Y. Yang, "From image vector to matrix: A straightforward image projection technique—IMPCA vs. PCA," *Pattern Recognit.*, vol. 35, no. 9, pp. 1997–1999, 2002.

[207] J. Ren, X. Jiang, and J. Yuan, "A chi-squared-transformed subspace of LBP histogram for visual recognition," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1893–1904, Jun. 2015.
[208] N.-S. Vu and A. Caplier, "Enhanced patterns of oriented edge magnitudes for face recognition and image matching," *IEEE Trans. Image Process.*, vol. 21, no. 3, pp. 1352–1365, Mar. 2012.
[209] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets, and J. Weng, "Discriminant analysis of principal components for face recognition," in *Face Recognition*. New York, NY, USA: Springer, 1998, pp. 73–85.
[210] C. Liu and H. Wechsler, "Robust coding schemes for indexing and retrieval from large face databases," *IEEE Trans. Image Process.*, vol. 9, no. 1, pp. 132–137, Jan. 2000.
[211] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall, "Learning a Mahalanobis metric from equivalence constraints," *J. Mach. Learn. Res.*, vol. 6, no. 1, pp. 937–965, 2006.
[212] W. R. Schwartz, H. Guo, J. Choi, and L. S. Davis, "Face identification using large feature sets," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 2245–2255, Apr. 2012.
[213] D. Chen, X. Cao, L. Wang, F. Wen, and J. Sun, "Bayesian face revisited: A joint formulation," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2012, pp. 566–579.
[214] X. Cao, D. Wipf, F. Wen, G. Duan, and J. Sun, "A practical transfer learning algorithm for face verification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3208–3215.
[215] C. Shan and T. Gritti, "Learning discriminative LBP-histogram bins for facial expression recognition," in *Proc. BMVC*, 2008, pp. 1–10.
[216] H. Méndez-Vázquez, Y. Martínez-Díaz, and Z. Chai, "Volume structured ordinal features with background similarity measure for video face recognition," in *Proc. Int. Conf. Biometrics (ICB)*, Jun. 2013, pp. 1–6.
[217] X. Wang, C. Zhang, and Z. Zhang, "Boosted multi-task learning for face verification with applications to Web image and video search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2009, pp. 142–149.
[218] T. Berg and P. N. Belhumeur, "Tom-vs-pete classifiers and identity-preserving alignment for face verification," in *Proc. BMVC*, vol. 2. 2012, p. 7.
[219] Y. Guo, G. Zhao, and M. Pietikäinen, "Discriminative features for texture description," *Pattern Recognit.*, vol. 45, no. 10, pp. 3834–3843, 2012.
[220] B. O'Connor and K. Roy, "Facial recognition using modified local binary pattern and random forest," *Int. J. Artif. Intell. Appl.*, vol. 4, no. 6, pp. 25–33, 2013.
[221] B. Jun, T. Kim, and D. Kim, "A compact local binary pattern using maximization of mutual information for face analysis," *Pattern Recognit.*, vol. 44, no. 3, pp. 532–543, Mar. 2011.
[222] J. Zhao, H. Wang, H. Ren, and S.-C. Kee, "LBP discriminant analysis for face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.-Workshops (CVPR Workshops)*, Sep. 2005, p. 167.
[223] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 1962–1977, Oct. 2011.
[224] Q. Yin, X. Tang, and J. Sun, "An associate-predict model for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 497–504.
[225] Y. Taigman, L. Wolf, and T. Hassner, "Multiple one-shots for utilizing class label information," in *Proc. BMVC*, 2009, pp. 1–12.
[226] L. Wolf, T. Hassner, and Y. Taigman, "Similarity scores based on background samples," in *Computer Vision—ACCV*. New York, NY, USA: Springer, 2010, pp. 88–97.
[227] C.-H. Chan, J. Kittler, and M. A. Tahir, "Kernel fusion of multiple histogram descriptors for robust face recognition," in *Structural, Syntactic, and Statistical Pattern Recognition*. New York, NY, USA: Springer, 2010, pp. 718–727.
[228] C. H. Chan, M. A. Tahir, J. Kittler, and M. Pietikäinen, "Multiscale local phase quantization for robust component-based face recognition using kernel fusion of multiple descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1164–1177, May 2013.
[229] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Sparse representation with kernels," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 423–434, Feb. 2013.
[230] B. He *et al.*, "Fast face recognition via sparse coding and extreme learning machine," *Cognit. Comput.*, vol. 6, no. 2, pp. 264–277, 2014.
[231] J. Qian and J. Yang, "General regression and representation model for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2013, pp. 166–172.
[232] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multitask joint sparse representation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.
[233] H. Zhang, N. M. Nasrabadi, Y. Zhang, and T. S. Huang, "Joint dynamic sparse representation for multi-view face recognition," *Pattern Recognit.*, vol. 45, no. 4, pp. 1290–1298, Apr. 2012.
[234] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 372–386, Feb. 2012.
[235] D.-S. Pham and S. Venkatesh, "Joint learning and dictionary construction for pattern recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.
[236] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2691–2698.
[237] L. Xu, X. Wu, K. Chen, and L. Yao, "Supervised within-class-similar discriminative dictionary learning for face recognition," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 561–572, Jul. 2016.
[238] X. Wu, Q. Li, L. Xu, K. Chen, and L. Yao, "Multi-feature kernel discriminant dictionary learning for face recognition," *Pattern Recognit.*, vol. 66, pp. 404–411, Jun. 2017.
[239] R. Timofte and L. Van Gool, "Adaptive and weighted collaborative representations for image classification," *Pattern Recognit. Lett.*, vol. 43, no. 7, pp. 127–135, Aug. 2014.
[240] J. Wu, R. Timofte, and L. Van Gool, "Learned collaborative representations for image classification," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, 2015, pp. 456–463.
[241] J. Yang, L. Zhang, Y. Xu, and J.-Y. Yang, "Beyond sparsity: The role of L_1-optimizer in pattern classification," *Pattern Recognit.*, vol. 45, no. 3, pp. 1104–1118, 2012.
[242] X. Peng, L. Zhang, Z. Yi, and K. K. Tan, "Learning locality-constrained collaborative representation for robust face recognition," *Pattern Recognit.*, vol. 47, no. 9, pp. 2794–2806, 2014.
[243] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 625–632.
[244] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2011, pp. 543–550.
[245] Z. Jiang, Z. Lin, and L. S. Davis, "Learning a discriminative dictionary for sparse coding via label consistent K-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1697–1704.
[246] W. Deng, J. Hu, and J. Guo, "In defense of sparsity based face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 399–406.
[247] C.-P. Wei and Y.-C. F. Wang, "Learning auxiliary dictionaries for undersampled face recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2013, pp. 1–6.
[248] M. Yang, L. Van Gool, and L. Zhang, "Sparse variation dictionary learning for face recognition with a single training sample per person," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 689–696.
[249] C.-P. Wei and Y.-C. F. Wang, "Undersampled face recognition via robust auxiliary dictionary learning," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1722–1734, Jun. 2015.
[250] V. M. Patel, T. Wu, S. Biswas, P. J. Phillips, and R. Chellappa, "Dictionary-based face recognition under variable lighting and pose," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 3, pp. 954–965, Jun. 2012.
[251] L. Ma, C. Wang, B. Xiao, and W. Zhou, "Sparse representation for face recognition based on discriminative low-rank dictionary learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2586–2593.
[252] M. Harandi, C. Sanderson, C. Shen, and B. C. Lovell, "Dictionary learning and sparse coding on grassmann manifolds: An extrinsic solution," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 3120–3127.
[253] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," *Large-Scale Kernel Mach.*, vol. 34, no. 5, pp. 1–41, 2007.
[254] R. Pascanu, Y. N. Dauphin, S. Ganguli, and Y. Bengio. (May 2014). "On the saddle point problem for non-convex optimization." [Online]. Available: https://arxiv.org/abs/1405.4604
[255] S. Chopra, S. Balakrishnan, and R. Gopalan, "DLID: Deep learning for domain adaptation by interpolating between domains," in *Proc. ICML Workshop Challenges Represent. Learn.*, vol. 2. 2013, pp. 1–8.

[256] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 513–520.

[257] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Computer Vision—ECCV*. Berlin, Germany: Springer, 2014, pp. 94–108.

[258] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. 27th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[259] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[260] G. B. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 2518–2525.

[261] D. Yi, Z. Lei, S. Liao, and S. Z. Li. (Jun. 2014). "Shared representation learning for heterogeneous face recognition." [Online]. Available: https://arxiv.org/abs/1406.1247

[262] Q. Liao, J. Z. Leibo, Y. Mroueh, and T. Poggio. (Nov. 2013). "Can a biologically-plausible hierarchy effectively replace face detection, alignment, and recognition pipelines?" [Online]. Available: https://arxiv.org/abs/1311.4082

[263] J. Lu, V. E. Liong, G. Wang, and P. Moulin, "Joint feature learning for face recognition," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 7, pp. 1371–1383, Jul. 2015.

[264] Y. Sun, X. Wang, and X. Tang, "Hybrid deep learning for face verification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 1489–1496.

[265] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. 27th Int. Conf. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[266] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1891–1898.

[267] H. Fan, Z. Cao, Y. Jiang, Q. Yin, and C. Doudou. (Mar. 2014). "Learning deep face representation." [Online]. Available: https://arxiv.org/abs/1403.2802

[268] M. Yang, X. Wang, G. Zeng, and L. Shen, "Joint and collaborative representation with local adaptive convolution feature for face recognition with single sample per person," *Pattern Recognit.*, vol. 66, pp. 117–128, Jun. 2017.

[269] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1701–1708.

[270] A. RoyChowdhury, T.-Y. Lin, S. Maji, and E. Learned-Miller. (Jun. 2015). "One-to-many face recognition with bilinear CNNs." [Online]. Available: https://arxiv.org/abs/1506.01342

[271] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[272] Y. Zhang, M. Shao, E. K. Wong, and Y. Fu, "Random faces guided sparse many-to-one encoder for pose-invariant face recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2416–2423.

[273] W. Wang, Z. Cui, H. Chang, S. Shan, and X. Chen. (Feb. 2014). "Deeply coupled auto-encoder networks for cross-view classification." [Online]. Available: https://arxiv.org/abs/1402.2031

[274] M. Kan, S. Shan, H. Chang, and X. Chen, "Stacked progressive auto-encoders (SPAE) for face recognition across poses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1883–1890.

[275] D. Cox and N. Pinto, "Beyond simple features: A large-scale feature search approach to unconstrained face recognition," in *Proc. IEEE Int. Conf. Autom. Face Gesture Recognit. Workshops (FG)*, Mar. 2011, pp. 8–15.

[276] E. Zhou, Z. Cao, and Q. Yin. (Jan. 2015). "Naive-deep face recognition: Touching the limit of LFW benchmark or not?" [Online]. Available: https://arxiv.org/abs/1501.04690

[277] Z. Zhu, P. Luo, X. Wang, and X. Tang. (Apr. 2014). "Recover canonical-view faces in the wild with deep neural networks." [Online]. Available: https://arxiv.org/abs/1404.3543

[278] J. Yim, H. Jung, B. Yoo, C. Choi, D. Park, and J. Kim, "Rotating your face using multi-task deep neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 676–684.

[279] C. Ding and D. Tao, "Robust face recognition via multimodal deep face representation," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 2049–2058, Nov. 2015.

[280] C. Ding and D. Tao, "Trunk-branch ensemble convolutional neural networks for video-based face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.

[281] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang. (Jun. 2015). "Targeting ultimate accuracy: Face recognition via deep embedding." [Online]. Available: https://arxiv.org/abs/1506.07310

[282] Y. Sun, X. Wang, and X. Tang. (Dec. 2014). "Deeply learned face representations are sparse, selective, and robust." [Online]. Available: https://arxiv.org/abs/1412.1265

[283] F. Schroff, D. Kalenichenko, and J. Philbin. (Mar. 2015). "FaceNet: A unified embedding for face recognition and clustering." [Online]. Available: https://arxiv.org/abs/1503.03832

[284] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[285] C. Szegedy *et al.* (Sep. 2014). "Going deeper with convolutions." [Online]. Available: https://arxiv.org/abs/1409.4842

[286] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, pp. 1–10, Dec. 2013. [Online]. Available: http://arxiv.org/abs/1312.4400

[287] Y. Sun, D. Liang, X. Wang, and X. Tang. (Feb. 2015). "DeepID3: Face recognition with very deep neural networks." [Online]. Available: https://arxiv.org/abs/1502.00873

[288] X. Wu, R. He, and Z. Sun. (Nov. 2015). "A light CNN for deep face representation with noisy labels." [Online]. Available: https://arxiv.org/abs/1511.02683

[289] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Deep learning identity-preserving face space," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 113–120.

[290] Z. Zhu, P. Luo, X. Wang, and X. Tang, "Multi-view perceptron: A deep model for learning face identity and view representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 217–225.

[291] Z. Zhu, P. Luo, X. Wang, and X. Tang. (Jun. 2014). "Deep learning multi-view representation for face recognition." [Online]. Available: https://arxiv.org/abs/1406.6947

[292] G. B. Huang and E. Learned-Miller, "Labeled faces in the wild: Updates and new reporting procedures," Dept. Comput. Sci., Univ. Massachusetts Amherst, Amherst, MA, USA, Tech. Rep. UM-CS-2014-003, 2014.

[293] J. Lu, Y.-P. Tan, and G. Wang, "Discriminative multimanifold analysis for face recognition from a single training sample per person," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 39–51, Jan. 2013.

HONGJUN WANG received the master's degree from the School of Information and Communication Engineering, Beijing University of Post and Telecommunications. His research interests include machine learning, computer vision, and information extraction.

JIANI HU received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China. He is currently an Associate Professor with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications. Her current research interests include computer vision, pattern recognition, and machine learning.

WEIHONG DENG received the Ph.D. degree from the Beijing University of Posts and Telecommunications, China. He is currently an Associate Professor with the Pattern Recognition and Intelligent Systems Laboratory, Beijing University of Posts and Telecommunications. His current research interests include computer vision, pattern recognition, and machine learning.