Rahman *et al. EURASIP Journal on Image and Video Processing* (2017) 2017:30
DOI 10.1186/s13640-017-0178-1

EURASIP Journal on Image and Video Processing

RESEARCH Open Access

DTCTH: a discriminative local pattern descriptor for image classification

Md. Mostafijur Rahman*, Shanto Rahman, Rayhanur Rahman, B. M. Mainul Hossain and Mohammad Shoyaib

Abstract

Despite lots of effort being exerted in designing feature descriptors, it is still challenging to find generalized feature descriptors, with acceptable discrimination ability, which are able to capture prominent features in various image processing applications. To address this issue, we propose a computationally feasible discriminative ternary census transform histogram (DTCTH) for image representation which uses dynamic thresholds to perceive the key properties of a feature descriptor. The code produced by DTCTH is more stable against intensity fluctuation, and it mainly captures the discriminative structural properties of an image by suppressing unnecessary background information. Thus, DTCTH becomes more generalized to be used in different applications with reasonable accuracies. To validate the generalizability of DTCTH, we have conducted rigorous experiments on five different applications considering nine benchmark datasets. The experimental results demonstrate that DTCTH performs as high as 28.08% better than the existing state-of-the-art feature descriptors such as GIST, SIFT, HOG, LBP, CLBP, OC-LBP, LGP, LTP, LAID, and CENTRIST.

Keywords: Discrimination ability, Event classification, Expression recognition, Image classification, Leaf classification, Noise adaptive, Object recognition, Scene classification, Ternary pattern

1 Introduction

Image classification has recently gained importance because of its numerous applications in different areas of image processing and computer vision such as texture classification [1–4], object tracking and recognition [5–9], scene classification [5, 7, 10–12], face detection and recognition [13–17], facial expression recognition [17–19], gender classification [17, 20], content-based image retrieval [21], and many others. These applications can be incorporated in video surveillance [22], human computer interaction [23], video and image retrieval [24], biometrics [25], and medical imaging [26–28].

Research works in this domain can be grouped into four different categories namely low-level, mid-level, high-level feature representations and classification strategies [29]. Among these, low-level feature representation plays a significant role since it is the building block for other steps. Therefore, many feature descriptors have been proposed for low-level feature representation. Among these, gradient [10, 30–32] and local binary pattern (LBP) [5, 7, 33] based methods are widely explored and proved to be successful in different applications. However, in most of the cases, these descriptors solve a particular problem and fail for general purpose image classification and/or consume high computational cost. To mitigate these problems, in this paper, we intend to develop a computationally low-cost general purpose feature descriptor that can perform well in diversified applications. The major challenge is that the real world applications are usually affected by large intra-class and small inter-class variations due to noise, illumination, photometric, scale, rotation, pose, and appearance variations [7]. Therefore, it becomes crucial to design a discriminative and robust feature descriptor which will address these issues.

Scale invariant feature transform (SIFT), histogram of oriented gradient (HOG), and GIST are the most commonly used gradient based low-level feature descriptors for image classification [9, 10, 30–32, 34]. Several extensions of SIFT such as speed up robust features (SURF) [35], gradient location and orientation histogram (GLOH) [36], and PCA-SIFT [37] have been introduced for improving classification accuracy and/or reducing computational complexity. Besides SIFT, HOG obtains both the properties of SIFT and GLOH [31]. Recently, an

*Correspondence: bit0312@iit.du.ac.bd
Institute of Information Technology, University of Dhaka, Dhaka-1000 Dhaka, Bangladesh

extension of HOG, namely histogram of second order gradient (HSOG) has been proposed to capture curvature information [9]. These descriptors usually use the first derivatives of an image (i.e., gradient direction and magnitude), which can capture local shape properties of the objects.

Gradient-based methods, such as SIFT first generally determines the salient points of an image and then calculate the descriptor on those points. The identification of salient points helps to capture the best discriminative foreground and discard the unnecessary background information. However, the identification of salient points is not directly incorporated to these descriptors. Moreover, in most of the cases, these methods do not consider the impact of human visual perception. Further, the gradient-based features often fail to distinguish between two pixels with same gradients even though those gradients correspond to different local structures [38].

In addition to the gradient-based methods, LBP and its extensions such as PRICoLBP [8], DDLBP [39], and OC-LBP [40] have become prominent because of their simplicity and better accuracy [41]. However, LBP-based methods that use "0" threshold have several major drawbacks such as,

1. Small changes in intensities due to noises in uniform and near-uniform regions often lead to wrong LBP codes. For example, in Fig. 1b, original intensity "154" (see Fig. 1a) is changed to "158," where LBP produces two different patterns (i.e., "11101000" and "11101100") though these two textures are similar.
2. LBP-based techniques fail to differentiate between the small and large differences in intensities, and these also fail to separate the foreground and background which degrades the discriminative ability. For example, differences between the center pixel ("170") and all of its eight neighboring pixels in Fig. 2a are small and in Fig. 2b are large except one pixel (i.e., "171"), whereas LBP encodes these two textures as same pattern (i.e., "11111111") which is not desired.

a

157	160	163
154	155	157
154	151	152

b

157	160	163
154	155	157
158	151	152

Fig. 1 Noise caused by local intensity fluctuation. **a** Original texture. **b** Texture changed due to local intensity fluctuation

a

171	174	175
173	170	172
174	175	171

b

190	195	194
182	170	171
193	197	183

Fig. 2 Example of two different textures which are encoded as same pattern by LBP, **a** small and **b** large differences

In LBP-based methods, all codes are calculated considering the center pixel and hence it can be considered as a background pixel in the local scope. Thus, all of its neighbors similar to it should also be considered as background pixels. Since the center pixel is "170," in Fig. 2b, the intensity "171" should be considered as a background and all other seven neighbors as foreground. However, LBP and most of its variants fail to achieve such discrimination ability.

A similar method to LBP is census transform (CT) [4]. Recently, CENTRIST is proposed for scene classification which uses CT of the image pixels [7]. However, due to the use of static threshold, CENTRIST has similar drawbacks like LBP. In order to address these issues (i.e., to extract the prominent features from an image and to deal with the presence of different levels of noises), few dynamic threshold-based methods are introduced. Local Gradient Pattern (LGP) is one of those which can adapt with local intensity fluctuations by considering mean of the local neighboring differences as a threshold [16]. However, LGP fails to differentiate between a positive and a negative change in the local neighborhoods due to providing same binary code (i.e., "1") in these two different directions. This problem can be solved by using ternary pattern [2, 3] which creates three patterns instead of two. Among the ternary pattern-based techniques, Local Ternary Pattern (LTP) shows resistance to the noises up to a certain level since it assumes that noises in an image usually vary within a fixed threshold ("±5") [2]. However, such a fixed threshold will not work for different types of images [3, 42].

To solve this issue, several dynamic threshold based methods are proposed such as noise tolerant ternary pattern (NTTP) and local adaptive image descriptor (LAID). However, the adaptive noise band defined in NTTP is application specific. Again in LAID, the median of the local neighboring differences is used as a threshold to generate the code. However, considering the median as a

threshold for a general purpose texture description might not be useful in many cases, because median cannot guarantee the proper separation of significant and insignificant changes since it is determined as the midpoint of data. Furthermore, despite the use of median as a threshold, it may have similar drawbacks like LBP, i.e., there might be a case when it will fail to adapt with intensity fluctuation (e.g., produces two different codes "01100011" and "01100111" for the texture in Fig. 1) and cannot discriminate between small and large intensity changes (e.g., produces same code "01100110" for two different textures in Fig. 2).

The incorporation of a non-zero threshold with LBP and its variants usually helps to reduce the effect of noise, suppress the background, and highlight the foreground. The benefit of such a threshold can further be realized by taking Weber's constant [43] into account. As per the Weber's law, it is not possible for human to distinguish the difference of intensities below the Weber's constant with naked eyes. Unfortunately, it is not easy to determine such a non-zero threshold that satisfies all of these issues. Hence, the desirable properties of a better threshold is that it will be able to (i) distinguish foreground and background, (ii) adapt with noise and other lighting conditions, and (iii) consistent with human visual perception.

In this paper, we introduce a new feature descriptor namely discriminative ternary census transform histogram (DTCTH) for general purpose image description. The threshold is determined for DTCTH in such a way so that it holds all the desirable properties and can be calculated in linear time. Further, a spatial pyramid representation is used with DTCTH for capturing the global structure of an image. The major contributions of this paper are summarized as follows.

1. We propose a dynamic threshold to produce stable code against intensity fluctuation.
2. The threshold can be calculated in linear time while it preserves all the desirable properties as mentioned above by utilizing only the center pixel. This threshold also helps to separate foreground and background of an image and complies with human visual perception.
3. The proposed DTCTH captures highly discriminative features by suppressing the fine details. Besides, the ternary code is generated to enhance the discrimination ability. We also incorporate a spatial pyramid representation which helps to boost the accuracy.
4. We show the generalizability of DTCTH in case of five different applications such as object, scene, event, leaf, and facial expression classification using nine standard datasets.

The rest of the paper is organized as follows. Section 2 briefly discusses existing state of the art low-level feature descriptors. Section 3 describes the use of these feature descriptors in different applications. The proposed method is described in Section 4. Section 5 presents a rigorous comparative experimental evaluation on five different applications. Section 6 concludes the paper with future research scope.

2 Background

A large number of techniques such as GIST, SIFT, HOG, LBP, CLBP, LGP, LTP, LAID, and CENTRIST have been proposed for image classification. These techniques capture texture patterns of an image. In this section, a brief description on all of these techniques are highlighted.

2.1 GIST

GIST descriptor is initially proposed in [10] where a low-dimensional representation of the scene is developed. The authors propose a set of perceptual dimensions (e.g., naturalness, openness, roughness, expansion, ruggedness) that represent the dominant spatial structure of a scene. The image is divided into small grids (e.g., 4×4 pixels), for which orientation histograms are extracted using 32 different Gabor filters at 4 scales and 8 orientations. Then the feature values within each grid are averaged. The final GIST descriptor is represented by combining the 16 averaged values of all scale and orientations, which results in $16 \times 32 = 512$ dimensions.

2.2 Scale invariant feature transform (SIFT)

Lowe et al. propose SIFT descriptor which consists of four major steps such as scale-space peak selection, keypoint localization, orientation assignment, and keypoint descriptor [30]. Firstly, potential interest points are identified in image over scale and space. This is implemented by constructing a Gaussian pyramid and searching for local peaks in a series of difference-of-Gaussian (DoG) images. Secondly, keypoints are localized to sub-pixel accurately by eliminating inconsistencies. Thirdly, the dominant orientations for each keypoint are identified based on the local image patch. Finally, a local image descriptor is produced for each keypoint, using the image gradients in the local neighborhood. In the representation of the descriptor, gradient locations are quantized into small location grids (e.g., 4×4 pixels), and the gradient directions are quantized into several (e.g., 8) orientations. SIFT descriptor is represented by combining histograms from all these small location grids (e.g., $4 \times 4 \times 8 = 128$ dimensions). To obtain illumination invariance, the descriptor is normalized by the square root of the sum of squared components.

2.3 Histogram of oriented gradient (HOG)

Dalal and Triggs introduce HOG descriptor which takes weighted votes depending on the gradient L2-norm for an orientated histogram channel [31]. HOG descriptor consists of several steps. The image is divided into small connected regions (e.g., 8×8 pixels) named as cells, and a histogram of gradient orientations is computed (e.g., using 1D centered derivative mask $[-1, 0, +1]$) for the pixels within each cell. Each cell is quantized into angular bins based on the gradient orientation. The pixels in each cell are used as a weighted gradient to the corresponding angular bin. The frequencies of histogram are also normalized using L2-norm to adapt with the variation of illumination. The final HOG descriptor is represented by combining these histograms.

2.4 Local binary pattern (LBP)

Ojala et al. first explore original LBP operator which thresholds $n \times n$ (e.g., 3×3) neighborhood of every pixel of an image with the center pixel value and considers the result as a binary number [1]. Each of the image pixel is then labeled with the corresponding decimal value of that binary number. The basic LBP is calculated using Eq. 1.

$$\begin{aligned} \mathrm{LBP}_{n,r}(x_c, y_c) &= \sum_{l=0}^{n-1} q(p_l - p_c) 2^l, \\ \text{Where } q(d) &= \begin{cases} 1, & \textit{if } d \geq 0 \\ 0, & \textit{otherwise} \end{cases} \end{aligned} \tag{1}$$

Here, n and r are the total number and the radius of the neighboring pixels. (x_c, y_c) is the coordinate of the center pixel c, p_l, and p_c are the intensities of the l^{th} neighboring and the center pixel (c) respectively. d is the difference between the neighboring and the center pixel. LBP codes can represent spatial micro-structures such as edge, corner, and line-end. Figure 3 presents some of these patterns.

LBP has 256 codes when eight neighbors are considered, which can be reduced to 59 codes by taking uniform patterns. The uniform patterns are calculated by Eq. 2.

$$\begin{aligned} U(\mathrm{LBP}_{n,r}(x_c, y_c)) &= |q(p_{n-1} - p_c) - q(p_0 - p_c)| \\ &\quad + \sum_{l=1}^{n-1} |q(p_l - p_c) - q(p_{l-1} - p_c)| \end{aligned} \tag{2}$$

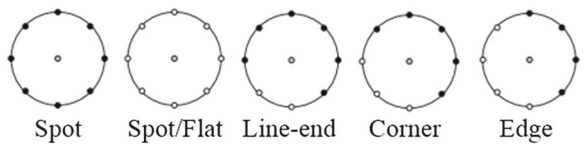

Fig. 3 Example of micro-structures encoded by LBP-based methods

2.5 Completed local binary pattern (CLBP)

Guo et al. [44] propose CLBP which consists of three components namely CLBP_S, CLBP_M, and CLBP_C. CLBP_S considers only the sign value of the differences between a pixel and its neighbors which is exactly same as LBP. CLBP_M uses the magnitudes of the differences between a pixel and its neighbors, and CLBP_C produces code by comparing the center pixel's intensity with the average image intensity. CLBP_M is generated following Eq. 3.

$$\begin{aligned} \mathrm{CLBP\_}M_{n,r}(x_c, y_c) &= \sum_{l=0}^{n-1} q(p_l - p_c) 2^l, \\ \text{Where } q(d) &= \begin{cases} 1, & \textit{if } d \geq T \\ 0, & \textit{otherwise} \end{cases} \end{aligned} \tag{3}$$

Here, T is the mean of all $|p_l - p_c|$ in the whole image. The CLBP_C is coded as Eq. 4.

$$\mathrm{CLBP\_}C(x_c, y_c) = q(p_c), q(d) = \begin{cases} 1, \textit{ if } d \geq T_I \\ 0, \textit{ otherwise} \end{cases} \tag{4}$$

Here, T_I is the average intensity of the whole image. These three operators (i.e., CLBP_C, CLBP_S, and CLBP_M) can be combined in two ways. The first way is to build a joint 3D histogram (CLBP_S/M/C), and the second one is to build a 2D joint histogram by combining CLBP_C with either CLBP_S (i.e., CLBP_S/C) or CLBP_M (i.e., CLBP_M/C). Then this 2D histogram is converted into a 1D histogram. Finally, CLBP_M_S/C or CLBP_S_M/C can be generated by concatenating CLBP_M with CLBP_S/C or CLBP_S with CLBP_M/C.

2.6 Local gradient pattern (LGP)

LGP is proposed by Jun et al. [16] where $n \times n$ (e.g., 3×3) neighborhood of a pixel is considered, and the neighbor having gradient greater than or equal to the average of gradients of eight neighboring pixels, is set to a binary value of "1", otherwise is assigned a binary value of "0", which is defined by Eq. 5.

$$\begin{aligned} \mathrm{LGP}_{n,r}(x_c, y_c) &= \sum_{l=0}^{n-1} q(g_l - g_\mu) 2^l, \\ \text{Where } q(d) &= \begin{cases} 1, & \textit{if } d \geq 0 \\ 0, & \textit{otherwise} \end{cases} \end{aligned} \tag{5}$$

Here, neighboring pixel and mean gradients are calculated as, $g_l = |p_l - p_c|$ and $g_\mu = \frac{1}{n} \sum_{l=0}^{n-1} g_l$ respectively where p_l and p_c are the neighboring and the center pixel's intensities.

2.7 Local ternary pattern (LTP)

Inspired by LBP, Tan and Triggs [2] introduce LTP operator. The key difference from LBP is the use of three bits to tackle intensity fluctuation instead of two bits in LBP.

Thus, LTP produces a ternary code which is calculated using Eq. 6.

$$\text{LTP}_{n,r}(x_c, y_c) = \sum_{l=0}^{n-1} q(p_l - p_c)3^l,$$
$$\text{Where } q(d) = \begin{cases} +1, & if\ d \geq\ 5 \\ -1, & if\ d \leq -5 \\ 0, & otherwise \end{cases} \quad (6)$$

Here, (x_c, y_c) is the coordinate of the center pixel *c*. p_c and p_l are the intensities of c and l^{th} neighboring pixels respectively. To reduce the size of the feature vector, a LTP code is usually split into two binary codes (i.e., upper and lower pattern) and these two types of codes are used for building two histograms separately. Finally, these two histograms are concatenated to represent the feature vector of an image.

2.8 Local adaptive image descriptor (LAID)

LAID is a recently proposed variant of LTP which uses a dynamic threshold to produce a ternary code. LAID operator is defined by Eq. 7.

$$\text{LAID}_{n,r}(x_c, y_c) = \sum_{l=0}^{n-1} q(p_l - p_c)3^l,$$
$$\text{Where } q(d) = \begin{cases} +1, & if\ d \geq\ T \\ -1, & if\ d \leq -T \\ 0, & otherwise \end{cases} \quad (7)$$

Here, (x_c, y_c) is the coordinate of the center pixel *c*. p_c and p_l are the intensities of c and l^{th} neighboring pixels respectively. T is a dynamic threshold which is determined by taking the median of $|p_l - p_c|$. Like LTP, a LAID code is split into two binary codes to reduce the size of the feature vector.

2.9 CENsus TRansform hISTogram (CENTRIST)

CENTRIST is a visual feature descriptor for scene and object classification which performs a census transform (CT) of an image and replaces the image with its CT values [7]. CT is a non-parametric local transformation designed for establishing relationships between local patches [4], which is calculated like LBP. CENTRIST does not use interpolation of corner pixels which is used in LBP. This is the only difference between LBP and CT calculation. The histogram of CT values has been computed to represent the visual descriptor. As CT only encodes the local structures of an image, CENTRIST uses the overlapped spatial pyramid to capture the global structures of an image in large scale. Finally, histograms of all blocks are concatenated to form the feature vector for classification.

3 Literature review

Till date, SIFT [30] is one of the most successful descriptors in different image processing applications such as scene and object classification. However, one of its major drawbacks is computational cost. Tola et al. propose DAISY descriptor which achieves computational gain by convolving orientation maps using Gaussian kernel [45]. They have used circular regions instead of regular grids where the radius is proportional to the standard deviation of the Gaussian kernel. Comparing different types of spatial pooling scheme, Brown et al. conclude that DAISY style pooling shows better accuracy while keeping lower computational cost [46]. Histogram of second order gradient (HSOG) adopts DAISY pooling, which at first computes a set of first order gradient maps (OGM), then second order gradient is calculated over all OGMs [9], resulted in the increase of both computational cost and accuracy.

SIFT and its variants can capture salient features using key-point descriptors [47], while HOG and its variants use magnitude as weight to deteremine the significance level of saliency in a particular direction. These processes can differentiate background and foreground information implicitly. However, in both cases, the computational cost could have been reduced, if the basic descriptor itself were able to identify the salient regions. Besides, most of these methods do not consider human visual perception to distinguish between background and foreground information. Moreover, a gradient-based method may fail to differentiate two different textures having the same gradient direction [38].

LBP and its variants [2, 13, 14, 17–20, 41] can capture local microstructures exploring different types of thresholds. These methods are commonly used for different applications such as face detection [15, 16], human detection [38], object, scene, event [48], face [13, 14], gender [20], and facial expression recognition [18, 49] for their convincing accuracy and lower computational cost. In most of the cases, an image is divided into several blocks where LBP-like codes are calculated and then histogram of these codes are calculated for each of these blocks. Finally, these histograms are concatenated to form the final feature vector. A similar but effective variant of this process is described in [18] where Shan et al. use LBP for facial expression recognition adopting boosted SVM. However, the basic LBP only uses sign information. Recently, CLBP is proposed which combines the sign and magnitude to extract more useful information [44] because the combination of sign and magnitude components can provide better clues, which are not evident if only a single component is considered individually [21]. Zhu et al. [40] propose orthogonal combination of local binary pattern (OC-LBP) which reduces the dimensionality of the basic LBP from 2^P to $4 \times P$. Due to considering four orthogonal neighbors for each OC-LBP code, this method fails to capture prominent textures even compared to LBP. However, the classification performance is boosted by incorporating bag of features with dictionary learning which increases computational cost. A recent

variant of LBP is local direction number pattern (LDN) [50], which performs well in face and expression recognition. LDN encodes the structure of a local neighborhood by analyzing its directional information. Consequently, LDN computes the edge responses in the neighborhood in eight different directions with a compass mask which also introduce extra computational burden.

Recently, Ren et al. have proposed data-driven LBP (DDLBP) for low-level image representation, which is formulated as a point selection problem, that is solved by maximal joint mutual information criterion [39]. This problem is converted into a binary quadratic programming problem and solved efficiently via the branch and bound algorithm. Hussain et al. address that existing local pattern descriptors using hand-specified coding limits those to small spatial supports and coarse gray-level comparisons and introduce local quantized pattern (LQP) which uses lookup table-based vector quantization to code larger or deeper patterns [51]. LQP inherits some of the flexibility and power of visual-word representations, without sacrificing the speed and simplicity of existing local patterns.

Inspired by the LBP and its variants, several ternary pattern-based methods such as LTP [2], NTTP [3], and LAID [42] are also introduced. In NTTP [3], the authors define an adaptive noise band to handle the influence of noise and use two types of thresholds for two different types of intensity regions. For low-intensity region, a constant threshold "τ" is used. However, defining the low-intensity region is not trivial. Again, τ needs to be set for a particular application and can vary from application to application. Such a setup might work for a particular application and thus it is necessary to find a proper threshold that can be used in general. LAID [42] is a recently explored local ternary pattern for texture classification which uses median of the local neighboring differences as a threshold. However, it may be affected by the non-linear property of median. For example, the median of [0, 1, 1, 2, 3, 4, 17, 18] is 2 or 3, as a result small differences (e.g., 3, 4) and large differences (e.g., 17, 18) will get the same code which is not expected. Such a scenario (also the opposite scenario [1, 2, 15, 16, 17, 17, 18, 19]) may commonly occur in many applications and thus results in inconsistent code. Hence, LAID may perform well for a particular application but might not be applicable in general.

Different from LBP, Gabor wavelet feature [52, 53] is one of the major approaches in terms of generality and performance in facial expression recognition. Gu et al. exploit Gabor feature for facial expression recognition which extends the radial encoding strategy for Gabor features based on retinotopic mapping that helps to obtain salient local features for facial expression representation [53]. Another feature descriptor using wavelet theory is distinctive efficient robust features (DERF) which utilizes exponential scale distribution, exponential grid structure, and circularly symmetric function difference of Gaussian as convolutional kernel [54]. DERF outperforms SIFT, HOG, and DAISY. However, Gabor-based methods and DERF are quite expensive in terms of computational cost.

On top of the basic features, there are few approaches which are used for mid- or high-level image representation [55–58]. Among these, Li et al. propose a high-level image representation named as object bank (OB) which describes an image as a scale-invariant response map of a large number of pre-trained generic object detectors [55]. Deformable part-based model (DPM) is introduced by Pandey and Lazebnik which uses latent SVM for classifying object and scene categories [56]. Besides these, Yang et al. propose spatial pyramid co-occurrence (SPCK++), which calculates spatial co-occurrences of visual words in a hierarchical spatial partitioning [57]. SPCK++ captures both the absolute and relative structure of an image by combining local co-occurrences with global partitioning. Image-to-class (I2C) distance is first used in NBNN [59] for image classification, which needs higher computational cost for nearest neighbor search in the testing phase. Recently, Wang et al. improve the discrimination of I2C distance especially for small number of local features by learning per-class Mahalanobis metrics [58].

For high-level representation, sparse coding-based approaches have shown better performance in image classification which usually adopt SIFT for low-level feature extraction. One of the first successful techniques is ScSPM [60] which uses sparse coding instead of vector quantization of SIFT descriptors. This technique adopts spatial max pooling (MP) of ScSPM features in regular SIFT grids for final feature representation. ScSPM performs better than both linear SPM kernel (LSPM) on histograms and traditional nonlinear SPM kernels with linear SVM (LSVM) because the pooling of sparse codes quantizes only the essential features which is linearly separable by SVM. However, ScSPM solves L1-norm optimization problem which is computationally expensive [61]. Moreover, it is non-consistent to encode similar descriptors [61, 62]. Several modifications have been proposed for these problems [61–63]. For instance, Wang et al. propose a modification of ScSPM by considering locality constraints in linear coding (LLC) to project each descriptor into its local-coordinate system where projected coordinates are amalgamated by MP [62]. Moreover, ScSPM, LLC, and most of the other sparse coding-based methods suffer from a severe drawback, which is the quantization of similar local features into different visual words [63]. To mitigate this problem, Oliveira et al. introduce sparse spatial coding (SSC) for image classification which combines a sparse coding dictionary learning, a spatial constraint coding, and an online classification stage [63]. The authors represent the final feature vector by adopting MP in SSC

features. Most of the sparse coding techniques [60, 62] are adopted on local features independently which consider the global similarity by constraint sparsity. However, dense local features share some local contextual information which is discarded by the existing sparse coding-based techniques and become less reliable when adopting spatial pooling [61]. To address this problem, a locality-constrained and spatially regularized (LCSR) coding is proposed by considering local spatial context of an image into the usual coding strategies which preserves locality constraints both in the feature space and spatial domain of the image [61]. The information loss in the feature quantization through pooling is still found, though several coding methods are introduced to address this problem. Wang et al. use linear distance coding (LDC) to alleviate this problem, which is a complementary technique to the traditional sparse coding schemes [64]. In their approach, local features of an image are transformed into discriminative distance vectors and then encodes these distance vectors into sparse codes to capture the salient features of the image.

Motivated by the sparse coding-based approaches, Gao et al. propose kernel sparse representation for image classification which performs sparse coding in a high-dimensional feature space mapped by implicit mapping function [65]. Afterwards, by combining these features with SPM, the authors propose Kernel Sparse Representation Spatial Pyramid Matching (KSRSPM). Besides this approach, Gao et al. [66] explore another sparse coding-based approach (LScSPM) by considering the instable sparse code produced by different sparse coding techniques [60, 62]. The authors use Laplacian sparse coding framework to address this issue. To reduce the high computational cost of dense kernel descriptors, efficient match kernel (EMK) is introduced which maps local features to a low-dimensional feature space and average the resulting vectors to form a set-level feature [67].

Apart from sparse coding-based methods, several approaches use soft-assignment coding [12, 68]. For example, Gemert et al. [12] introduce soft-assignment of codewords using kernel density estimation which assigns local features to all the visual codewords [68]. Comparing with other existing coding schemes, soft-assignment coding is simple and has low computational cost. However, the major drawback is that it cannot produce comparable result with other coding schemes [68]. Liu et al. address that the inferiority of soft-assignment coding is because of its negligence to the underlying manifold structure of local features and propose a localized soft-assignment coding (LSA) [68]. They use mix-order max pooling (MMP) instead of general MP which helps to boost the performance.

Along with the aforementioned supervised learning techniques, several unsupervised learning techniques are also used in computer vision. For example, Bosch et al. [34] introduce a semi-supervised learning (SP-pLSA) by combining the unsupervised probabilistic latent semantic analysis (pLSA) [69] and a discriminative classifier for image classification. Here, pLSA is applied to the images which are represented by the frequency of visual words where color SIFT is used as a basic descriptor. Recently, deep learning-based unsupervised technique of feature learning is adopted that does not require manual intervention. This approach has gained popularity because of its better accuracy. Using multiple levels of representation and abstraction, it helps a machine to understand about data (e.g., images, sound, and text) more accurately. In deep learning frameworks, first, unsupervised feature learning is performed on a large image dataset and then the weights of the deep network is adjusted. Eventually, a model is built that can later be used to solve a particular problem which is known as fine tuning. Among the existing popular models, AlexNet [70], Places-CNN [71], and VGG_S [72] are widely used because they cover diversified applications. Despite the gain of popularity of deep learning, it is very computation intensive and requires expensive hardware and large set of training data. Furthermore, a well-defined network structure is also required to solve a particular set of problems which is challenging and usually fix up empirically.

The mid- or high-level feature representation aims to capture strong spatial layouts, encodes salient textures, and makes those working with linear classifier [56, 60, 62]. To achieve the aforementioned properties, these methods incorporate different steps such as generative part models [59, 73], discriminative codebook learning [68, 74], sparse coding [60, 62, 66], and/or spatial pooling [62]. The incorporation of these steps lead to increase in computational cost. However, if it is possible to incorporate these issues to the basic feature descriptor, it may reduce the huge computational cost of the mid-/high-level representation.

Apart from these levels (low, mid, and high) of representations, classifiers also play an important role in classification accuracies, such as SVM with different kernels (e.g., linear, polynomial and RBF kernel) are used for classification in various applications [7, 17, 18, 50]. In general, although RBF kernel produces better results in many applications, its computational cost is high. A fast and effective classification is thus necessary which can be achieved in two ways such as by selecting relevant features where nonlinear relationship of features is already incorporated and then use LSVM, or by introducing a low-cost nonlinear kernel of SVM. Maji et al. [75] introduce a fast non-linear kernel of SVM namely histogram intersection kernel which achieves better classification accuracy in many applications [76]. Zhang et al. propose a hybrid classification technique (SVM-KNN) which selects features using k nearest neighbors [77] and

classify using DAGSVM [78] classifier. SVM-KNN has low computational cost and performs well when the test image is similar to one of the training images. However, this technique fails to generalize much beyond the labeled images because of calculating image-to-image distance. Recently, to perform fast and better classification, Jianxin Wu [79] introduces PmSVM, which solves a dual SVM formulation using a coordinate descent approach. PmSVM approximates the gradient using polynomial regression instead of the kernel function and feature mapping.

From the above discussion, it can be seen that most of the existing techniques attempt to capture the salient textures that are stable against different lighting conditions, noises, intensity fluctuations which help to clearly represent necessary foreground information. For this purpose, these approaches either include preprocessing such as keypoint identification before generating descriptor or postprocessing such as different costly high-level representations. However, the computational cost of these approaches can be reduced if it is possible to identify the prominent features using only the basic low-level descriptors. Therefore, it is desirable to come up with a mechanism that can identify prominent features in a low-level descriptor.

4 Proposed method

In this paper, we propose a new feature descriptor named as discriminative ternary census transform histogram (DTCTH) for image representation which holds most of the key properties of a feature descriptor. The overall process of the construction of descriptor is described in the following subsections.

4.1 Desired properties

A feature descriptor for image classification should have the following essential properties.

1. *Discrimination ability*: A feature descriptor should have higher discrimination ability. If it has the capability to encode only the class-specific information by suppressing the unnecessary background, it will perform well in image classification. Figure 4 presents several images with corresponding Sobel images from different object, scene, and expression classes. All of these images contain respective class-specific information which is clearly visualized from their Sobel images. This class-specific information needs to be encoded for better image classification. Therefore, our goal is to encode only this class-specific information by discarding the unnecessary background details.
2. *Illumination invariance*: A good feature descriptor should be able to adapt with illumination changes because illumination of same image can vary due to different reasons. Among the existing low-level feature descriptors, CENTRIST-based methods have this property and if we follow the basic CENTRIST structure, our proposed descriptor will have the same property.
3. *Generalizability*: It is expected that a descriptor has reasonable accuracy for different types of applications. This can be achieved when a descriptor is capable to encode class-specific features and suppress unnecessary background information for the respective applications. We will design our descriptor such a way that it will have this property.
4. *Incorporation of visual perception*: In general, a person cannot distinguish a change in an image if the change is below the Weber's constant [43]. So, it is reasonable to conclude that the changes below this constant is not necessary to capture. Thus, a descriptor should have the capability to capture only those changes that is important for human vision.
5. *Stable code*: Producing stable code (i.e., same code) against intensity fluctuation is another essential property for a feature descriptor. It is obvious that intensity of an image might be changed for several reasons. Let $\delta = p_l - p_c$, where p_c is the intensity of a target pixel c and p_l is the intensity of its l^{th} neighbor. If the difference of intensities $\mid \delta \mid$, of the two pixel is large, those two pixels should be considered differently and vice versa. Hence, the range of δ has to be set in such a way so that the two pixels get the same or different codes in two different situations. At this point, we define two terms *certain* and *uncertain* state for a code (C) using Eq. 8.

$$C = \begin{cases} \text{certain state}, & if \ \mid \delta \mid \geq T \\ \text{uncertain state}, & otherwise \end{cases} \tag{8}$$

Here, T is a threshold that might be static or dynamic. Defining certain and uncertain states have several advantageous properties. For example, in this case, we can achieve discriminative and stable code because of considering the certain and uncertain states separately. Apart from that, we can get three groups (G) of codes using Eq. 9. Group one ($g1$) and group three ($g3$) belong to certain state, while group two ($g2$) remains in uncertain state.

$$G = \begin{cases} g1, \ if \ \delta \geq T \\ g2, \ if \ -T < \delta < T \\ g3, \ if \ \delta \leq -T \end{cases} \tag{9}$$

Again T should be dynamic because a static threshold might fail in case of different types of images.

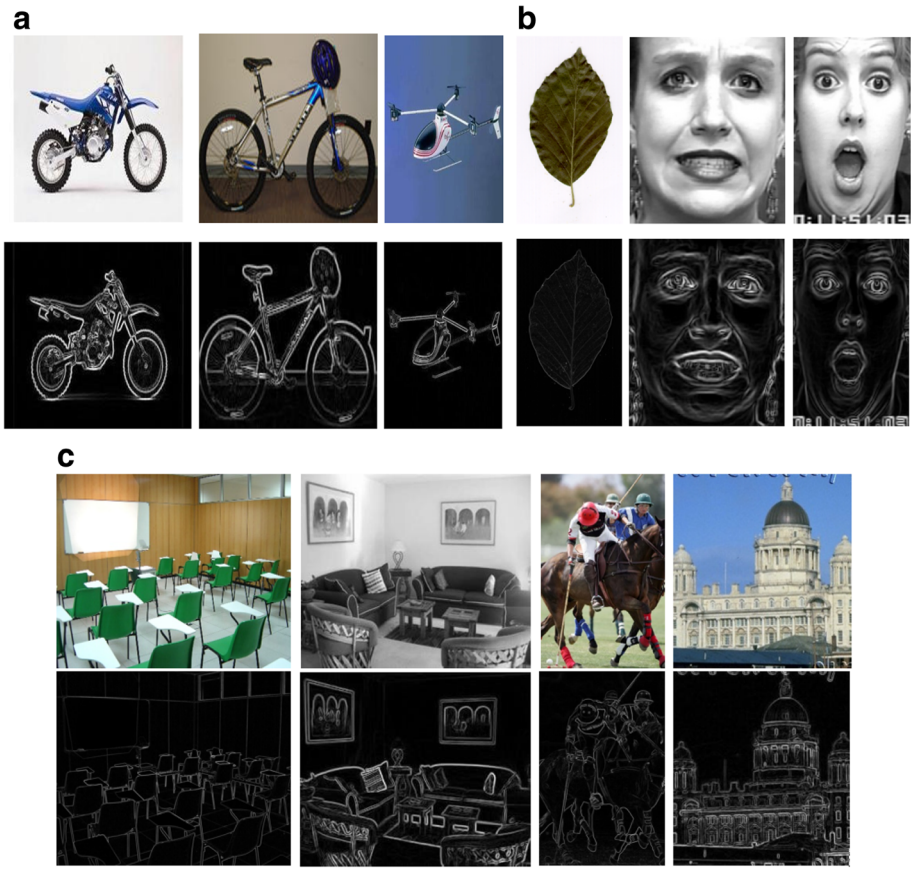

Fig. 4 Sample images with corresponding Sobel images from different categories of object, scene, and expression (first row original images and second row Sobel images). **a** Object classes. **b** Leaf and expression classes. **c** Scene classes

4.2 Discriminative ternary census transform histogram (DTCTH)

The overall process of DTCTH calculation is shown in Fig. 5. For producing different codes for certain and uncertain changes of intensities in an image, we consider ternary coding scheme, namely discriminative census transform (DCT) which is calculated using Eq. 10.

$$\mathrm{DCT}_{n,r}(x_c,y_c) = \sum_{l=0}^{n-1} q(p_l - p_c)3^l,$$
$$\text{Where } q(d) = \begin{cases} +1, & if\ \ d \geq T \\ -1, & if\ \ d \leq -T \\ 0, & otherwise \end{cases} \tag{10}$$

Here, T is a dynamic threshold, n and r are the total number of neighbors and the radius of the neighboring pixels respectively, and (x_c, y_c) is the coordinate of the center pixel. p_c and p_l are the intensities of the center pixel c and l^{th} neighboring pixel. For simplicity and computational efficiency, the ternary pattern is divided into two census transformed images namely upper (DCT_UP) and lower (DCT_LP) pattern which are calculated using Eqs. 11 and 12. Figure 6 shows a pictorial example of DCT calculation. Afterwards, two separate histograms such as H_DCTUP and H_DCTLP of these two binary patterns are calculated using Eqs. 13 and 14. The final feature vector is represented by concatenating these histograms.

$$DCT\_UP_{n,r}(x_c,y_c) = \sum_{l=0}^{n-1} q(p_l - p_c)2^l,$$
$$\text{Where } q(d) = \begin{cases} 1, & if\ \ d \geq T \\ 0, & otherwise \end{cases} \tag{11}$$

$$DCT\_LP_{n,r}(x_c,y_c) = \sum_{l=0}^{n-1} q(p_l - p_c)2^l,$$
$$\text{Where } q(d) = \begin{cases} 1, & if\ \ d \leq -T \\ 0, & otherwise \end{cases} \tag{12}$$

$$H\_DCTUP^k = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \delta^k_{DCT\_UP_{n,r}(i,j)},$$
$$\text{Where } \delta^k_p = \begin{cases} 1, & if\ \ p = k \\ 0, & otherwise \end{cases} \tag{13}$$

$$H\_DCTLP^k = \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} \delta^k_{DCT\_LP_{n,r}(i,j)},$$
$$\text{Where } \delta^k_p = \begin{cases} 1, & if\ \ p = k \\ 0, & otherwise \end{cases} \tag{14}$$

Here, $DCT\_UP_{n,r}(i,j)$ and $DCT\_LP_{n,r}(i,j)$ are the upper and lower DCT codes of coordinate (i, j). k is the k^{th} bin

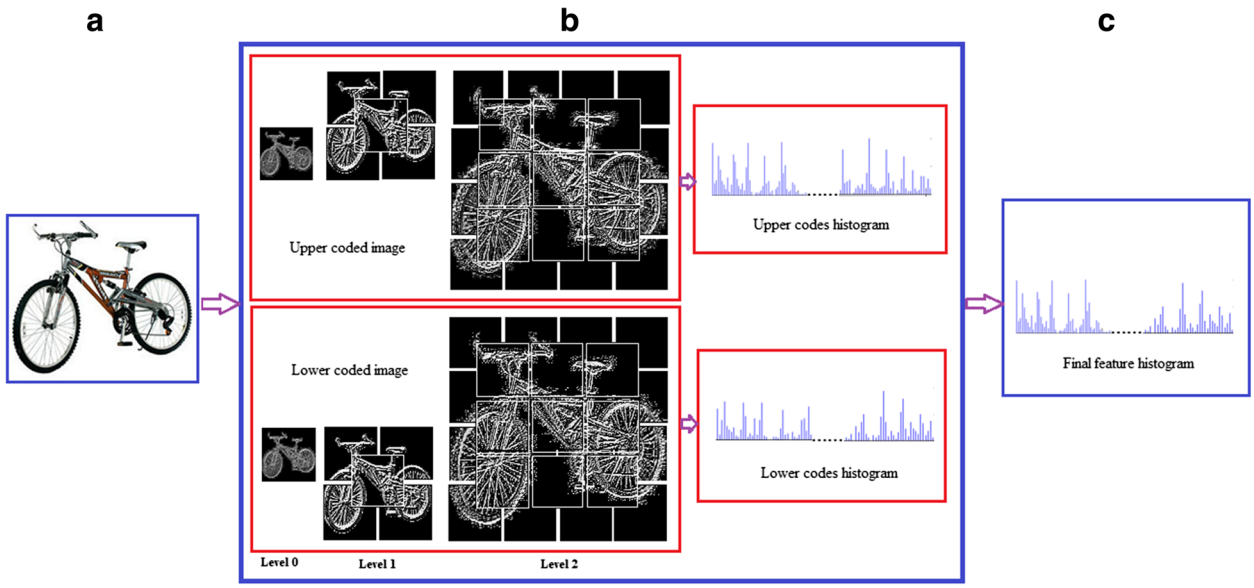

Fig. 5 Overall process of DTCTH calculation. **a** Input image. **b** DTCTH codes in overlapping SPM. **c** Final feature histogram

of the histogram. h and w are the height and width of the image block. Kronecker delta δ_p^k is a piecewise function of p and k. As DCT encodes only local micro-structures of spatial location, spatial pyramid representation (SPM) which is used in CENTRIST, is adopted to capture the global structures of an image.

4.3 Determining the value of *T*

During the calculation of DTCTH, for every pixel of an image, we have eight different values that are obtained by calculating the differences ($|\ \delta\ |$) from its neighbors. We want to partition these values into two groups using a threshold such that the variance is maximized between the groups and is minimized within the group. The purpose of this partitioning is that the group with lower values can be considered as background where the group with higher values as foreground, in the local scope. A solution of this partitioning problem can be found using Jenks natural breaks optimization method [80, 81]. However, such an optimization is very time consuming as we have to apply the method for each pixel of an image to generate the respective code. Furthermore, Jenks natural breaks optimization is not designed to comply with Weber's constant though the combination of these two is expected to increase the accuracy. Under these circumstances, after exhaustive empirical analysis (on 10^9 samples), we set the value of T to the square root of the center pixel in a local ternary pattern. We have found that such a choice of T brings about the closest possible similarity which is around 84.90%, to the aforementioned optimization problem considering Weber's constant. Thus, we can conclude that taking the square root of the center pixel is a very close approximation of the desired threshold with much low computational cost.

For validating the value of T, we have performed rigorous experiment on four different applications with

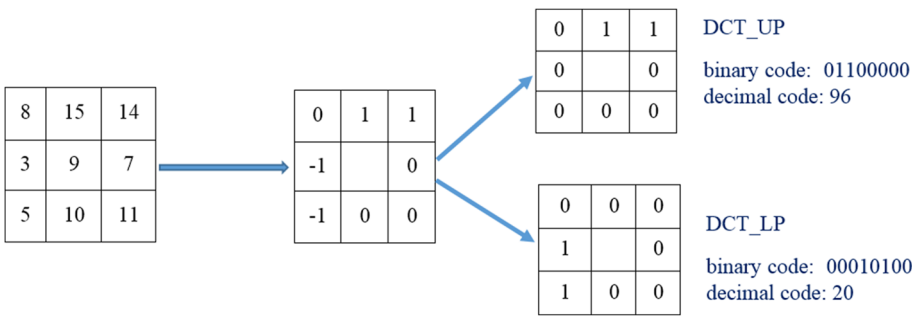

Fig. 6 An example of DCT calculation ($T = 3$)

four datasets using different values of T. The dataset includes Caltech-101 [73] (102 classes and 9,145 images) for object classification, UIUC Sport Event [33] (8 classes and 1586 images) for event classification, OT scene [10] (8 classes and 2688 images) for scene classification, and Cohn Kanade [82] (6 classes and 960 images) for expression recognition. We consider both the fixed and dynamic thresholds to determine the value of T. From the experiments, we observe that the accuracy is decreased for the values greater than 20, and hence, we consider the values up to 25, both in fixed and dynamic cases. The mean and median of the differences among the neighboring pixels and the center pixel, SQRT, and cube root of the center pixel are also considered. Figure 7a shows the accuracy of different fixed thresholds and square root threshold, and Fig. 7b illustrates the accuracy of different dynamic thresholds, as mentioned earlier. From Fig. 7, it is found that T is defined as SQRT of center pixel and performs best for all applications. Using *McNemar's* test, we observe that the proposed SQRT threshold resulted in significantly fewer mis-classifications than other thresholds (maximum P value, $P = 0.001$ and minimum P value, $P = 3.83932E - 28$).

4.4 Properties of DTCTH

DTCTH encodes micro-structures such as line, edge, and corners which are stable against intensity fluctuation and monotonic illumination variation. Some of these properties are described in the following.

DTCTH captures more relevant part of an image that are necessary for recognizing an object/scene. To understand this, let us consider Fig. 4 which includes examples from three different applications. Now, if we only have the Sobel images where all fine details are suppressed and only the class-specific information is retained then it will help a classifier to achieve better accuracy. Likewise, if we analyze the images in Fig. 8, we can easily find that DTCTH suppresses most of the background information keeping the necessary details compared to the others. As the proposed technique have this property, it is more generalized compared to other descriptors.

DTCTH features are more robust to noise, and it produces stable code by adapting the intensity fluctuations in local neighborhood. For example, CENTRIST and LBP fail to produce the same code in case of intensity fluctuation (see Fig. 1), whereas DTCTH is successful in this case (i.e., "00000000"). Furthermore, we add white Gaussian noise to the original images as shown in Fig. 9 to test the robustness to noise of DTCTH. Now, if we compare the coded images with or without noises for DTCTH and CENTRIST, we can easily find that DTCTH is more robust to noise and thus can capture the face specific feature by eliminating the details.

Besides these, for certain intensity changes in positive and negative directions, DTCTH produces two different codes for these two directions which is desired because from this type of representation, we can get more detailed information about the local micro-structure of an image. For example, in Fig. 10, DTCTH produces three different codes for aforementioned three groups of codes following Eq. 9 such as uncertain state (i.e., 0 for 71 and 69), intensity changes in positive direction (i.e., 1 for 80, 81 and 79), and negative direction (i.e., −1 for 61, 60, and 60) in certain regions by considering 70 as the center pixel.

To understand the effect of human visual perception in case of DTCTH, we require a reference value to measure the change in intensity. Since DTCTH uses the center pixel for calculating code, we use the same reference point for measuring Weber's constant. For example, in Fig. 10, the Weber's constant for the neighboring pixels 71 and 69

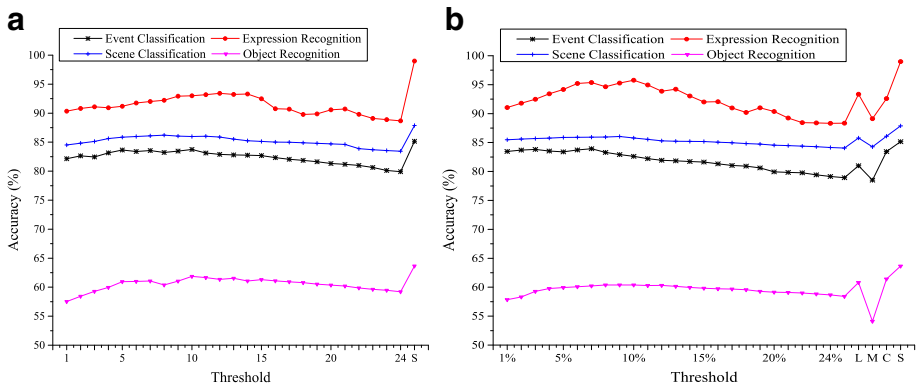

Fig. 7 Demonstration of object recognition, scene classification, event classification, and expression recognition accuracy using different thresholds. **a** Accuracy of different constant thresholds. **b** Accuracy of different dynamic thresholds (i.e., percentage of the center pixel as threshold). All the results are generated considering ternary pattern in CENTRIST framework. Noteworthy, *L* (threshold used in LGP [16]), *M* (median threshold used in LAID [42]), *S* (SQRT of center pixel), and *C* (cube root of center pixel)

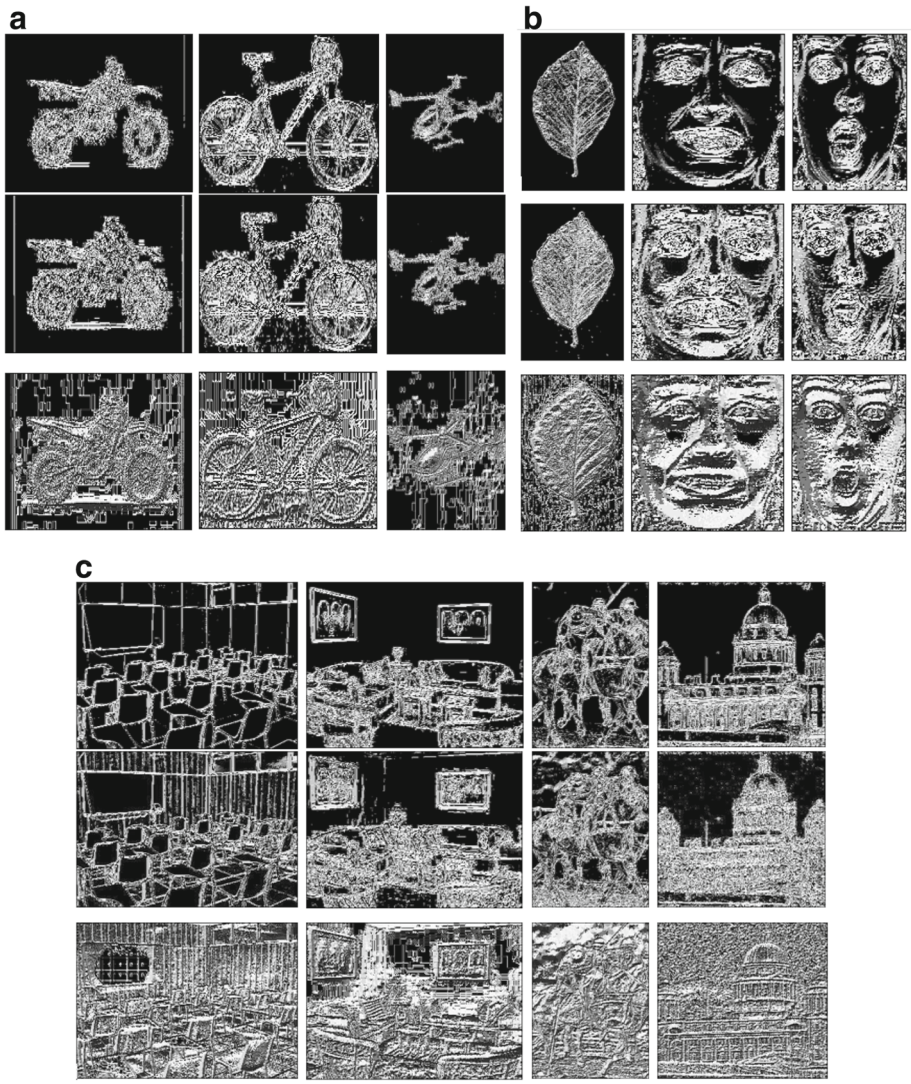

Fig. 8 Coded image by DTCTH, LTP, and CENTRIST for corresponding images in Fig. 4. **a** Object classes. **b** Leaf and expression classes. **c** Scene classes (*first row* —DTCTH-coded image, *second row*—LTP-coded image, and *third row*—CENTRIST-coded image)

with the center pixel 70 is 0.01, which is below the Weber's constant for human visual perception [43]. Hence, these two neighbors are coded as "0." Similarly, 79 and 61 have the Weber's constant of 0.13; hence, these two pixels are coded as "1" and "-1" due to the changes in two different directions which is expected. From Fig. 11, it is understandable that the smooth regions (e.g., cheek area) are coded similarly and the codes for $+ve$ and $-ve$ directions contain complementary information (e.g., eyebrow and mouth regions).

5 Experimental evaluation

In this section, we evaluate the performance of DTCTH by comparing with the other state-of-the-art methods over nine datasets that belong to five different applications such as object, scene, event, leaf, and facial expression classification. Application wise descriptions of the experiments are discussed in the following.

Table 1 presents the overview of these nine datasets along with the number of training and testing samples used in the experiments, which is also described in the respective datasets. For the experiments, all images are resized to at most 300×300 pixels. Except the expression recognition, the dataset is split into five random partitions and experiments are performed five times. That is, we have performed fivefold cross validation and report the average accuracies in the respective tables. In case of expression recognition, the experiments are run ten times with person independent splits by following the standard protocol, and the average accuracies are reported in the

Fig. 9 Noise resistance of CENTRIST vs. DTCTH. **a** Image without noise. **b** Noisy image (*middle* one is CENTRIST-coded image, *right* one is DTCTH-coded image)

tables. The datasets description, followed by the proper comparison with state-of-the-art methods, are described in details in the following subsections. In this paper, we also provide results of some of the deep learning-based techniques for completeness, though these techniques are not directly comparable to DTCTH.

For implementing DTCTH, few parameters are related to the basic descriptor (DTCTH) and its classifier (SVM). The major parameters for DTCTH are its radius (r) and its number of neighbors (n). From the literature, we have found that the best accuracies (with reasonable feature vector length) are produced using $r = 1$ or $r = 2$ and

80	81	79
71	70	69
61	60	60

Fig. 10 Illustrative example of the certain and uncertain regions in an image

$n = 8$ in most of the applications [13]. For SVM, different types of kernels such as linear, RBF, polynomial, sigmoid, and histogram intersection (HI) can be used. For first four kernels, we use LibSVM package[1]. To find out how DTCTH behaves with these parameters, we use three datasets namely Caltech 101, UIUC Sports Event, and Scene 15. The results with these parameters' settings for these datasets are summarized in Table 2 which shows that DTCTH works well in most of the cases when $n = 8$ neighbors at radius $r = 2$ is considered with HI kernel.

In this work, we mainly adopt the CENTRIST framework[2], keeping all the parameters same as described in CENTRIST [7]. Thus, for fair comparison, we consider *eight* neighbors at radius *one* from the center pixel-like CENTRIST in all the experiments, although consideration of other parameter setting may produce better result. Following CENTRIST, we also avoid corner points interpolation and remove two DCT bins (i.e., 0 and 255) while calculating DCT histograms. Afterwards, we take the square root of DTCTH histogram and perform L1 normalization of those descriptors. For classification, we use SVM classifier with linear kernel ($c = 2^{-5}$, $g = 2^{-7}$) [83] and histogram intersection (HI) kernel [75]. We use the aforementioned parameter settings unless otherwise stated. To reflect a brief description of a particular method, we mainly consider the following representation for Tables 3–11. Firstly, we give the basic descriptor name followed by mid-/high-level representation in the parentheses, then the classifier name and publication year.

5.1 Object classification

We have considered two well-known and most challenging object datasets named as Caltech-101 [73] and Caltech-256 [11] to evaluate the object recognition performance of the proposed descriptor. These two datasets are described below followed by the obtained results from the experiments.

5.1.1 Caltech-101

Caltech-101 contains 9144 images of 101 categories and an additional background category, making a total number of 102 categories, with significant variance in shape [73]. The number of images per category varies from 31 to 800. As suggested by the original dataset [73] and many other researchers [5–7, 60], we have partitioned the whole dataset into 5, 10, 15, 20, 25, and 30 training images per class and rest for testing to measure the performance unless otherwise stated.

To compute DTCTH code, it only compares its pixel values with a specific value (square root of the center pixel) and performs better than SIFT, DAISY, and HSOG techniques. DTCTH achieves 78.56% accurate object classification rate by considering only the low-level feature representation, which demonstrates the improvement of

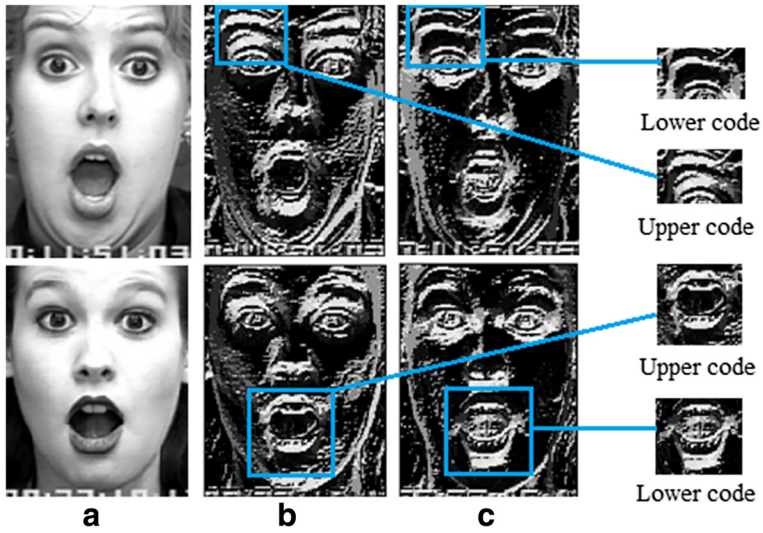

Fig. 11 DTCTH-coded image. **a** Input image. **b** Upper code. **c** Lower code

performance over existing state-of-the-art methods such as SSC [63], ScSPM [60], LSPM [60], LLC [62], LCSR [61], and LDC [64], even though most of these methods use different high-level representations. A recent state-of-the-art method namely Gaussian of local descriptors (GOLD) [85] that achieves 80.92% accuracy using 30 training and only 50 testing samples. It uses dense color SIFT as a basic descriptor and focuses on high-level representation. The result is comparable when we use DTCTH with $r = 2$. However, the computational cost of this method is much higher compared to us. Colored SIFT (CSIFT) [86] is another recent state-of-the-art low-level descriptor that also uses LLC as a high-level representation. However, this method produces inferior result (69.18%) compared to DTCTH.

Apart from the aforementioned techniques, other well-known descriptors such as GIST [10], CENTRIST [7], LTP [2], and LGP [16] are used in different applications and compared with DTCTH. For the sake of fair comparison, the results of CENTRIST, LTP, and LGP are generated using same parameter settings that we have used. The result of GIST descriptor is generated using the standard setup, which is 32 Gabor filters in 4 scales and 8 orientations. All of these low-level feature descriptors produce inferior results in comparison with DTCTH (see Table 3). It is observed from this table that PmSVM [79] performs (72.18%) slightly better than DTCTH (71.84%) considering 15 training images. It is noteworthy that they have used different classifier than ours and considered only 20 sample images for testing.

5.1.2 Caltech-256

Caltech-256 is a very challenging dataset which contains 30,607 images of 256 categories and an additional clutter category [11]. Each class has at least 80 images which show higher variability in object size, location, and pose than that in Caltech-101. We have evaluated our algorithm in different settings such as considering 15, 30, 45, and 60 training images per class and using the rest as test data unless otherwise stated.

Table 4 presents the experimental results of DTCTH as well as existing state-of-the-art methods in the literature on Caltech-256 dataset which shows that the proposed DTCTH performs better compared to other basic feature descriptors including GIST [10], CENTRIST [7], LTP [2], and LGP [16]. Besides Borji et al. [88] perform a

Table 1 Different benchmark datasets with proper training samples

Applications	Object classification		Event classification	Scene classification			Leaf classification	Facial expression classification	
Databases	Caltech-256	Caltech-101	UIUC sports event	OT scene	Scene 15	Indoor 67	Swedish leaf	Cohn Kanade (CK)	CK+
Classes	257	102	8	8	15	67	15	6/7	7
Total samples	30,608	9145	1586	2688	4485	5620	1125	960/1280	981
Training images/class	60	30	70	100	100	80	25	Person independent	
Test images/class	Rest	Rest	60	Rest	Rest	20	Rest	10-fold cross-validation	

Table 2 Effect of different SVM and DTCTH parameters on UIUC Sports Event, Caltech 101, and Scene 15 datasets

Techniques	UIUC sports event	Caltech 101	Scene 15
Linear kernel			
$DTCTH_{8,1}$	85.16±0.96	72.26±1.67	82.66±0.50
$DTCTH_{8,2}$	84.73±1.01	76.08±0.41	82.87±0.49
Polynomial kernel			
$DTCTH_{8,1}$	83.69±0.97	68.64±0.53	80.92±0.12
$DTCTH_{8,2}$	84.02±1.15	73.21±0.58	82.62±0.56
RBF kernel			
$DTCTH_{8,1}$	75.74±1.54	58.72±1.16	72.95±0.62
$DTCTH_{8,2}$	75.83±1.17	63.96±1.36	73.73±0.65
Sigmoid kernel			
$DTCTH_{8,1}$	67.95±1.94	52.59±1.41	68.16±1.88
$DTCTH_{8,2}$	70.47±1.58	56.88±1.31	69.59±1.08
Histogram intersection kernel			
$DTCTH_{8,1}$	88.18±0.84	78.56±0.91	83.63±0.21
$DTCTH_{8,2}$	87.75±0.57	80.36±0.24	83.92±0.43

Here, we consider 70 training and 60 test images for UIUC Sports Event, 30 training and remaining test images for Caltech 101, and 100 training and remaining test images for Scene 15

comparative evaluation of different existing techniques such as SIFT [88], HOG [88], HOG pyramid [88], LBP [88], and LBP pyramid [88] on this dataset, all of which produce inferior results compared to DTCTH. Moreover, DTCTH achieves more than 11 and 17% accuracy improvements over CENTRIST and GIST respectively by considering HI kernel.

Furthermore, DTCTH performs better than different sparse and soft-assignment coding-based approaches including ScSPM [60], KSRSPM [65], LScSPM [66], EMK [67], LSA [68], SSC [63], and LDC [64] except LLC [62]. This LLC shows slightly better result (47.68%) compared to DTCTH (45.61%) with the cost of high-level representation. It is noteworthy that such high-level representation is computationally expensive. In contrast, the proposed DTCTH achieves comparable accuracy with much lower computation. A recent state-of-the-art low-level descriptor is reversal invariant descriptor enhancement (RIDE) [89] which improves the performance of basic SIFT using a high-level representation that uses improved fisher vector (IFV) [90]. This IFV helps to boost up of the performance and achieves 60.25% accuracy.

5.2 Scene classification

We have implemented DTCTH for both indoor and outdoor scene classification. For this purpose, three datasets such as MIT Indoor 67 [91] for indoor, OT scene [10] for outdoor, and Scene 15 [5] for both indoor and outdoor scene classification are used. The description of these three datasets are discussed below followed by the experimental results.

MIT Indoor 67. This dataset holds 15,620 images of 67 indoor scene categories [91]. There are at least 100 images in each category. We randomly choose 80 images from each category for training and remaining images for testing the system.

OT scene. Oliva and Torralba at first used OT scene dataset for scene classification [10]. It consists of 2688 images from 8 scene classes. In the experiments, 100 images are randomly selected to train the system and the other images are used for testing purpose.

Scene 15. Scene 15 dataset contains 4485 images of 15 scene categories [5]. Each category has between 200 and 400 images. We randomly select 100 images from each category as training data and use the remaining images as test data.

In general, indoor scene classification is comparatively challenging than outdoor scene classification because indoor scenes contain large inter-class similarity. Therefore, the performance of all the methods are generally lower for indoor scene (e.g., MIT Indoor 67) compared to the outdoor scene (e.g., OT scene) datasets. Several state-of-the-art low-level feature descriptors such as PRI-CoLBP [8], CENTRIST [7], GIST [10], SIFT [30], HOG [31], HSOG [9], CS-LBP [87], LTP [2], and LGP [16] are explored for both indoor and outdoor scene classification. Recently, CENTRIST has been extended to multiple channels (mCENTRIST) [6], which shows better result (44.60%) in indoor scene classification than CENTRIST (35.12%). They have also showed that multi-channel GIST (mGIST) and multi-channel SIFT (mSIFT) perform better than original GIST and SIFT respectively. DTCTH obtains better accuracy than all of these approaches in all the datasets (see Tables 5, 6 and 7).

Besides these basic features, there are other methods such as NBNN [59], PmSVM [79], pLSA [69], SP-pLSA [34], Bag-of-Phrase (BoP) [95], and DAISY [45] which are also used for scene classification. To this end, DTCTH achieves better results in the respective datasets than most of these approaches. In few cases, such as SP-pLSA shows slightly better results (83.7%) considering color SIFT for Scene 15 dataset compared to DTCTH (83.63%). However, DTCTH achieves higher accuracy (89.18%) compared to SP-pLSA (87.80%) in OT scene dataset. BoP uses histogram mining with discriminative learning technique and achieves 86.78% accuracy in Scene 15 dataset. RIDE achieves 64.93% accuracy on MIT Indoor 67 dataset by adopting IFV which is computationally expensive as described earlier [89]. In OT scene dataset, DTCTH achieves the highest correct classification rate (89.18%). In this dataset, comparing with GIST which is designed for scene classification, DTCTH increases the performance

Table 3 Object classification rate (%) in Caltech-101

Techniques	5	10	15	20	25	30
Places-CNN, 2014 [71]	–	–	–	–	–	65.18
ImageNet-CNN, 2014 [71]	–	–	–	–	–	87.22
Hybride-CNN, 2014 [71]	–	–	–	–	–	84.79
Dense color SIFT (SP-pLSA) SVM, 2008 [34]	–	–	59.80 (50)*	-	-	67.70 (50)*
SIFT (ML + CORR) KNN, 2008 [84]	–	–	61.00	–	–	69.60
SIFT (ML + PMK) KNN, 2008 [84]	–	–	52.20	–	–	62.10
Dense SIFT (KC) SVM with HI, 2008 [12]	–	–	–	–	-	64.14 (50)*
Dense SIFT (LSPM + MP) LSVM, 2009 [60]	–	–	53.23	–	–	58.81
Dense SIFT (ScSPM + MP) LSVM, 2009 [60]	–	–	67.00	–	–	73.20
Dense SIFT (LLC + MP) LSVM, 2010 [62]	51.15	59.77	65.43	67.74	70.16	73.44
Dense SIFT (LSA + MMP) LSVM, 2011 [68]	–	–	–	–	–	74.21
Dense SIFT (LDC + LLC/LSA + MP) LSVM, 2013 [64]	–	–	–	–	–	74.47
Dense SIFT (LCSR + MP) LSVM, 2012 [61]	–	–	–	–	–	73.23
Dense color SIFT (GOLD) LSVM, 2015 [85]	–	–	73.39 (at most 50)*	–	–	80.92 (at most 50)*
Dense SIFT (SSC + MP) OCL, 2012 [63]	55.64	65.52	69.98	73.99	75.49	77.59
HSOG (LLC + MP) SVM, 2014 [9]	–	–	60.46 (15)*	–	–	67.97 (15)*
CSIFT (LLC + MP) LSVM, 2015 [86]	46.48	56.97	62.09	65.45	68.17	69.18
Dense SIFT (BoF) SVM, 2004 [9, 30]	–	–	62.48 (15)*	–	–	69.89 (15)*
$CS-LBP_{2,8,0.01}$ (BoF) SVM, 2009 [9, 87]	-	-	58.50 (15)*	–	–	66.86 (15)*
DAISY (BoF) SVM, 2010 [9, 45]	-	-	58.63 (15)*	–	–	67.01 (15)*
SIFT (SPM) SVM, 2006 [5]	–	–	56.40 (50)*	–	–	64.60
Dense SIFT (SPM) SVM, 2007 [11]	44.20	54.50	59.00	63.30	65.80	67.60
Dense SIFT + NBNN, 2008 [59]	–	–	65.00 (20)*	–	–	70.40
Geometric blur + SVM-KNN, 2006 [77]	46.60	55.80	59.05	62.00	–	66.23
Dense SIFT (BoF) PmSVM-χ^2, 2012 [79]	–	–	72.08 (20)*	–	–	–
Dense SIFT (BoF) PmSVM-HI, 2012 [79]	–	–	72.18 (20)*	–	–	–
LGP (SPM) LSVM, 2013	39.86	50.11	57.84	60.03	62.96	66.52
OC-LBP (BoF) LSVM, 2013	47.10	56.34	62.43	64.70	67.63	70.87
LAID (SPM) LSVM, 2013	39.03	48.35	54.11	57.83	60.84	63.87
CLBP_S/M/C (SPM) LSVM, 2010	32.06	40.03	45.59	49.40	52.56	55.35
LTP (SPM) LSVM, 2010	41.04	51.23	59.69	61.17	64.57	67.85
GIST + LSVM, 2001	40.16	47.87	52.5	56.25	58.88	61.70
CENTRIST (SPM) LSVM, 2011	39.46	49.72	55.84	59.47	62.25	65.23
Proposed (DTCTH + LSVM)	46.98	57.00	63.66	65.83	68.69	72.26
Proposed (DTCTH + HI)	56.74	65.97	71.84	74.80	76.85	78.56

*Different number of test images used for the experiment rather than standard settings

over 18 and 20% by considering linear and HI kernel respectively. DTCTH provides 83.63% accuracy in Scene 15 dataset which also demonstrates 2 and 28% improvements over CENTRIST and GIST respectively. Furthermore, DTCTH outperforms object bank, DPM, SPCK++, and NBNN in the respective datasets.

Considering high-level image representation, sparse and soft-assignment coding-based approaches are well-known. Among these approaches, ScSPM [60], LLC [62], SSC [63], LSA [68], LCSR [61] and LDC [64] have gained popularity for scene classification. Most of these approaches use two steps for feature representation such

Table 4 Object classification rate (%) in Caltech-256

Techniques	15	30	45	50	60
Places-CNN, 2014 [71]	–	–	–	–	45.59
ImageNet-CNN, 2014 [71]	–	–	–	–	67.23
Hybride-CNN, 2014 [71]	–	–	–	–	65.06
SIFT (SPM + pLSA) SVM, 2006 [5]	–	34.10	–	–	–
Dense SIFT (LSPM + MP) LSVM, 2009 [60]	13.20±0.62	15.45±0.37	16.37± 0.47	–	16.57±1.01
Dense SIFT (KSRSPM) LSVM, 2010 [65]	29.77±0.14	35.67±0.10	38.61±0.19	–	40.30±0.22
Dense SIFT (KC) SVM with HI, 2008 [12]	–	27.17 (25)*	–	–	–
Dense SIFT (EMK) LSVM, 2009 [67]	23.20±0.60	30.50±0.40	34.40±0.40		37.60±0.50
Dense SIFT (ScSPM + MP) LSVM, 2009 [60]	27.73±0.51	34.02±0.35	37.46±0.55	–	40.14±0.91
Dense SIFT (LLC + MP) LSVM, 2010 [62]	34.36	41.19	45.31	–	47.68
Dense SIFT (LSA + MMP) LSVM, 2011 [68]	–	–	–	–	36.52±0.26
Dense SIFT (LDC + LLC/LSA + MP) LSVM, 2013 [64]	–	–	–	–	38.25±0.08
Dense SIFT (LScSPM + MP) LSVM, 2013 [66]	29.99±0.15	35.74±0.10	38.47±0.51	–	40.32±0.32
Dense SIFT (SSC + MP) OCL, 2012 [63]	30.59±0.35	37.08±0.36	40.68±0.16	–	43.48±0.38
CSIFT (LLC + MP) LSVM, 2015 [86]	28.58±0.35	35.20±0.36	38.97±0.16	–	41.31±0.38
SIFT + SVM, 2004 [30, 88]	–	–	–	29.4	–
HOG + SVM, 2005 [31, 88]	–	–	–	33.3	–
HOG (SPM) SVM, 2014 [31, 88]	–	–	–	32.7	–
LBP + SVM, 2002 [1, 88]	–	–	–	20.7	–
LBP (SPM) SVM, 2014 [1, 88]	–	–	–	20.5	–
Dense SIFT (SPM) SVM, 2007 [11]	28.30	34.10	–	–	–
Dense SIFT + NBNN, 2008 [59]	30.4 (25)*	36.0 (25)*	–	–	–
LGP (SPM) LSVM, 2013	22.86±0.41	28.89±0.33	31.13±0.28	32.02±0.29	33.14±0.51
OC-LBP (BoF) LSVM, 2013	25.77±0.22	31.28±0.26	34.91±0.27	35.52±0.29	37.83±0.32
LAID (SPM) LSVM, 2013	19.71±0.33	25.45±0.36	29.08±0.29	30.25±0.44	32.65±0.4
CLBP_S/M/C (SPM) LSVM, 2010	15.72±0.17	20.48±0.34	24.08±0.35	25.16±0.45	27.56±0.5
LTP (SPM) LSVM, 2010	23.12±0.26	29.33±0.27	31.74±0.35	32.95±0.37	33.97±0.43
GIST + LSVM, 2001	18.58±0.27	21.36±0.15	24.17±0.12	26.14±0.29	27.09±0.5
CENTRIST (SPM) LSVM, 2011	21±0.34	27.13±0.29	29.97±0.31	31.12±0.43	32.72±0.82
Proposed (DTCTH + LSVM)	27.43±0.37	33.57±0.43	36.38±0.33	37.59±0.35	38.30±0.31
Proposed (DTCTH + HI)	32.91±0.31	39.42±0.21	43.07±0.18	44.16±0.25	45.61±0.27

*Different number of test images used for the experiment rather than standard settings

Table 5 Scene classification rate (%) in MIT Indoor 67

Techniques	Accuracy
CNN-SVM, 2014 [92]	58.4
Places-CNN, 2014 [71]	68.24
ImageNet-CNN, 2014 [71]	56.79
Hybride-CNN, 2014 [71]	70.80
Dense SIFT (LSA + MMP) LSVM, 2011 [68]	44.19
dense SIFT (LLC + MP) LSVM, 2010 [62]	43.78
dense SIFT (LDC + LLC/LSA + MP) LSVM, 2013 [64]	46.69
Dense SIFT (SSC + MP) OCL, 2012 [63]	44.35
Object Bank + LSVM, 2010 [55]	37.60
Dense SIFT (BoF) SVM with HI, 2014 [93]	45.86
DPM, 2011 [56]	30.40
CENTRIST (BoF) PmSVM-HI, 2012 [79]	47.15
CENTRIST (BoF) PmSVM-χ^2, 2012 [79]	46.20
PRICoLBP + SVM with χ^2, 2014 [8]	43.4
HOG, 2005 [56]	22.8
SPM, 2006 [8],	34.4
MM-scene, 2010 [94]	28.00
mCENTRIST (SPM) LSVM, 2014 [6]	44.6±1.2
mSIFT (SPM) LSVM, 2014 [6]	39.7±1.6
mGIST (SPM) LSVM, 2014 [6]	31.5±1.6
LGP (SPM) LSVM, 2013	34.24±1.12
OC-LBP (BoF) LSVM, 2013	36.99±2.34
LAID (SPM) LSVM, 2013	32.78±1.47
CLBP_S/M/C (SPM) LSVM, 2010	30.45±1.70
LTP (SPM) LSVM, 2010	35.87±1.23
GIST + LSVM, 2001	26.5±1.41
CENTRIST (SPM) LSVM, 2011	35.12±0.99
Proposed (DTCTH + LSVM)	43.33±0.72
Proposed (DTCTH + HI)	46.22±1.02

as feature encoding and pooling (e.g., average, max) steps. Boureau et al. [96] perform a comparative experimental analysis which shows that sparse coding with MP achieves better result than other combinations in Scene 15. Among all of these approaches, only LDC [64] achieves slightly better classification accuracy (46.69%) than DTCTH (46.22%) in MIT indoor 67, but this method produces inferior results compared to DTCTH in Caltech-101 (4.09% inferior), Caltech-256 (7.36% inferior), and Scene 15 (1.13% inferior) datasets.

Table 6 Scene classification rate (%) in OT scene

Techniques	Accuracy
Dense color SIFT (pLSA) KNN, 2006 [69]	86.65
Dense color SIFT (pLSA) SVM, 2008 [34]	82.50
Dense color SIFT (SP-pLSA) SVM, 2008 [34]	87.80
HSOG (LLC + MP) SVM, 2014 [9]	86.30 *
dense SIFT (BoF) SVM, 2004 [9, 30]	84.10 *
HOG (BoF) SVM, 2005 [9, 31]	82.40 *
DAISY (BoF) SVM, 2010 [9, 45]	85.70 *
$CS-LBP_{2,8,0.01}$ (BoF) SVM, 2009 [9, 87]	83.40 *
Dense color SIFT (SPM) SVM, 2008 [34]	87.10
LGP (SPM) LSVM, 2013	84.52
OC-LBP (BoF) LSVM, 2013	84.67
LAID (SPM) LSVM, 2013	84.25
CLBP_S/M/C (SPM) LSVM, 2010	79.34
LTP (SPM) LSVM, 2010	85.60
GIST + LSVM, 2001	69.03
CENTRIST (SPM) LSVM, 2011	84.01
Proposed (DTCTH + LSVM)	87.88±0.51
Proposed (DTCTH + HI)	89.18±0.81

*Half of the images for training and another half for testing

5.3 Event classification

The description of the dataset followed by experimental results are discussed in the following.

UIUC Sports Event. This dataset consists of 1579 images of 8 sports event categories [33]. The number of images in each class ranges from 137 to 250. We have followed the experimental settings described in [77] which is, randomly selecting 70 images as the training and other 60 for testing.

DTCTH (88.18%) outperforms all the low-level descriptors (as described before) even mCENTRIST [6] (86.50%) that uses color information for this dataset (see Table 8). It also shows better result compared to many high-level representation (see Table 8) with few exceptions such as BoP (91.74%) that uses saliency map and mining strategy to boost-up its performance [95].

5.4 Leaf classification

For leaf classification, we use Swedish leaf dataset [97]. The dataset description followed by experimental results are discussed in the following.

Swedish leaf. This dataset consists of 15 species of leaves with 75 images per species [97]. The dataset has two properties such as the leaf images are manually aligned well and in a good shape. Following the standard protocol discussed in [8], 25 randomly selected images from each species are used for training and the rest for testing.

Table 7 Scene classification rate (%) in Scene 15

Techniques	Accuracy
Places-CNN [71]	90.19
ImageNet-CNN [71]	84.23
Hybride-CNN [71]	91.59
SIFT (SPM + pLSA) SVM, 2006 [5]	81.40 ± 0.50
Dense color SIFT (pLSA) SVM, 2008 [34]	72.70
Dense color SIFT (SP-pLSA) SVM, 2008 [34]	83.70
Dense SIFT (KC) SVM with HI, 2008 [12]	77.10
Dense SIFT (LSPM + MP) LSVM, 2009 [60]	65.32 ± 1.02
Dense SIFT (ScSPM + MP) LSVM, 2009 [60]	80.28 ± 0.93
Dense SIFT (Sparse Code) LSVM, 2010 [96]	84.10 ± 0.50
Dense SIFT (LLC + MP) LSVM, 2010 [62, 64]	79.81 ± 0.35
Dense SIFT (LSA + MMP) LSVM, 2011 [68]	82.70 ± 0.39
SIFT (BOVW + SPCK++) SVM, 2011 [57]	82.51 ± 0.43
Dense SIFT (LDC + LLC/LSA + MP) LSVM, 2013 [64]	82.50 ± 0.47
Dense SIFT (LCSR + MP) LSVM, 2012 [61]	82.67 ± 0.51
Object bank + LSVM, 2010 [55]	80.90
Dense SIFT + I2CDML, 2010 [58]	77.00 ± 0.60
Dense SIFT (SPM) I2CDML, 2010 [58]	81.20 ± 0.52
Dense SIFT + NBNN, 2008 [58, 59]	72.30 ± 0.93
PRICoLBP + SVM with χ^2, 2014 [8]	82.04
Dense SIFT (BoF) SVM with HI, 2014 [93]	82.06
LGP (SPM) LSVM, 2013	78.22 ± 0.56
OC-LBP (BoF) LSVM, 2013	77.22 ± 0.40
LAID (SPM) LSVM, 2013	81.18 ± 0.60
CLBP_S/M/C (SPM) LSVM, 2010	76.47 ± 0.15
LTP (SPM) LSVM, 2010	80.25 ± 0.31
GIST + LSVM, 2001	55.55 ± 0.67
CENTRIST (SPM) LSVM, 2011	81.45 ± 0.23
Proposed (DTCTH + LSVM)	82.66 ± 0.50
Proposed (DTCTH + HI)	83.63 ± 0.21

Table 8 Event classification rate (%) in UIUC Sports Event

Techniques	Accuracy
Places-CNN, 2014 [71]	94.12
ImageNet-CNN, 2014 [71]	94.42
Hybride-CNN, 2014 [71]	94.22
Dense SIFT (KSRSPM) LSVM, 2010 [65]	84.92 ± 0.78
Dense SIFT (ScSPM + MP) LSVM, 2009 [60]	82.74 ± 1.46
Dense SIFT (LSA + MMP) LSVM, 2011 [68]	82.29 ± 1.84
Dense SIFT (LLC + MP) LSVM, 2010 [62]	81.41 ± 1.84
Dense SIFT (LCSR + MP) LSVM, 2012 [61]	87.23 ± 1.14
Dense SIFT + I2CDML, 2010 [58]	78.5 ± 1.63
Dense SIFT (SPM) I2CDML, 2010 [58]	79.7 ± 1.83
Dense SIFT + NBNN, 2008 [58, 59]	67.6 ± 1.1
Dense SIFT (BoF) SVM with HI, 2014 [30, 93]	85.12
LQP + SVM with RBF, 2012 [39, 51]	78.9
DDLBP + Max Relevance + SVM with RBF, 2014 [39]	83.5
DDLBP + mRMR + SVM with RBF, 2014 [39]	83.5
DDLBP + MJMI + SVM with RBF, 2014 [39]	84.0
mGIST (SPM) LSVM, 2014 [6]	76.2 ± 1.9
mSIFT (SPM) LSVM, 2014 [6]	84.2 ± 0.7
mCENTRIST (SPM) LSVM, 2014 [6]	86.5 ± 0.6
LGP (SPM) LSVM, 2013	78.42 ± 0.94
OC-LBP (BoF) LSVM, 2013	81.15 ± 2.18
LAID (SPM) LSVM, 2013	78.50 ± 0.65
CLBP_S/M/C (SPM) LSVM, 2010	78.88 ± 0.92
LTP (SPM) LSVM, 2010	82.43 ± 1.17
GIST + LSVM, 2001	69.95 ± 0.98
CENTRIST (SPM) LSVM, 2011	79.50 ± 0.95
Proposed (DTCTH + LSVM)	85.16 ± 0.96
Proposed (DTCTH + HI)	88.18 ± 0.84

Table 9 presents the experimental results of DTCTH as well as the existing techniques in literature on this dataset, which shows that DTCTH achieves 99.52% accuracy. Several techniques are used in this dataset for shape and leaf classification. DTCTH outperforms all of these approaches by considering gray-scale image as input which is provided in Table 9.

5.5 Facial expression recognition

We also evaluate the performance of DTCTH in expression recognition. Most of the facial expression recognition

Table 9 Leaf classification rate (%) in Swedish leaf

Techniques	Accuracy	Input
Soderkvist, 2001 [97]	82.40	Contour
SC + DP, 2007 [98]	88.12	Contour
IDSC + DP, 2007 [98]	94.13	Contour
SPTC + DP, 2007 [98]	95.33	Gray-scale
Shape-Tree, 2007 [99]	96.28	Contour
CENTRIST, 2011 [7, 8]	90.61	Contour
SLPA, 2013 [100]	96.33	Gray-scale
PRICoLBP + SVM with χ^2, 2014 [8]	99.38	Gray-scale
LGP (SPM) LSVM, 2013	98.08	Gray-scale
OC-LBP (BoF) LSVM, 2013	99.36	Gray-scale
LAID (SPM) LSVM, 2013	99.33	Gray-scale
CLBP_S/M/C (SPM) LSVM, 2010	98.53	Gray-scale
LTP (SPM) LSVM, 2010	98.20	Gray-scale
GIST + LSVM, 2001	96.08	Gray-scale
CENTRIST (SPM) LSVM, 2011	97.44	Gray-scale
Proposed (DTCTH + LSVM)	99.49	Gray-scale
Proposed (DTCTH + HI)	99.52	Gray-scale

systems attempt to recognize a set of expressions like anger, disgust, fear, joy, sadness, and surprise. This 6-class expression set can also be extended to a 7-class expression set including a neutral expression. In this work, our aim is to recognize both 6-class and 7-class expressions. For this purpose, we have performed experiments on Cohn Kanade (CK) [82] and CK+ [103] datasets, where person independent 10-fold cross-validation testing is considered. More specifically, the whole dataset is divided into ten person independent groups of roughly equal number of subjects. Nine groups are used to train the classifier, and the remaining group is used as the test data. The datasets description along with experimental results are discussed in the following.

CK and CK+ Dataset. The CK dataset consists of 100 university students who were between 18 and 30 years old at the time of their inclusion. Among them, 65% are female. In the experimental setup, 320 image sequences are selected from 96 subjects, each of which is labeled as one of the six basic expressions. For 6-class expression recognition, the three most expressive image frames are taken from each sequence that results in 960 expression images. In order to build the neutral expression set, the first frame (i.e., neutral expression) from all 320 sequences

Table 10 Expression recognition rate (%) in CK

Techniques	CK	
	6-class expression	7-class expression
Ranzato et al. [101]	–	90.10
LBP, 2006 [13]	92.60 ± 2.90	88.90 ± 3.50
LBP + Template Matching, 2009 [18]	84.50 ± 5.20	79.10 ± 4.60
Geometric feature + TAN, 2003 [102]	–	73.20
LBP + SVM, 2009 [18]	91.50 ± 3.10	88.10 ± 3.80
Boosted-LBP, 2009 [18]	89.80 ± 4.70	85.00 ± 4.50
Boosted-LBP + SVM, 2009 [18]	95.00 ± 3.20	91.10 ± 4.00
Gabor + SVM, 2003 [52]	–	84.80
Gabor, 2009 [18]	89.40 ± 3.00	86.60 ± 4.10
LDN + LSVM, 2013 [50]	98.40 ± 1.40	92.30 ± 3.00
LGP (SPM) LSVM, 2013	93.36 ± 3.76	88.97 ± 4.18
OC-LBP (BoF) LSVM, 2013	84.84 ± 5.29	78.17 ± 5.50
LAID (SPM) LSVM, 2013	89.13 ± 5.41	84.21 ± 4.73
CLBP_S/M/C (SPM) LSVM, 2010	85.44 ± 4.92	78.59 ± 5.78
LTP (SPM) LSVM, 2010	91.18 ± 8.68	88.79 ± 2.31
CENTRIST (SPM) LSVM, 2011	89.84 ± 7.90	86.69 ± 2.04
Proposed (DTCTH + LSVM)	98.98 ± 1.29	92.75 ± 5.43
Proposed (DTCTH + HI)	97.76 ± 2.43	93.89 ± 2.63

Table 11 7-class expression recognition rate (%) in CK+

Techniques	Accuracy
AUDN, 2013 [104]	92.05
SPTS, 2006 [69]	50.40
CAPP, 2006 [69]	66.70
SPTS + CAPP, 2006 [69]	83.30
LDN + LSVM, 2013 [50]	89.30
NABP + Adaboost, 2015 [17]	92.17
LBP + Adaboost, 2006 [17]	88.67
LTP + Adaboost, 2010 [17]	89.65
LGP + Adaboost, 2013 [17]	83.10
HOG + Adaboost, 2005 [17]	89.69
OC-LBP + BoF + LSVM, 2013	84.20 ± 4.90
LAID (SPM) LSVM, 2013	92.76
CLBP_S/M/C (SPM) LSVM, 2010	87.47
CENTRIST (SPM) LSVM, 2011	88.70 ± 4.37
Proposed (DTCTH + LSVM)	93.99 ± 5.83
Proposed (DTCTH + HI)	93.82 ± 5.52

Table 12 Confusion matrix of DTCTH in case of 6-class expression recognition on CK

	Anger	Disgust	Fear	Sadness	Happy	Surprise
Anger	99.22	0.0	0.78	0.0	0.0	0.0
Disgust	0.0	100.0	0.0	0.0	0.0	0.0
Fear	0.0	0.0	97.22	0.0	2.78	0.0
Sadness	0.83	0.0	0.0	98.33	0.0	0.83
Happy	0.43	0.0	0.85	0.0	98.72	0.0
Surprise	0.0	0.0	0.0	0.0	0.0	100.0

are selected to make the 7-class expression dataset (1280 images). Furthermore, the extended CK (CK+) is used, which includes 593 sequences for seven basic expressions including happiness, sadness, surprise, anger, disgust, fear, and contempt. In the experiments, we select the most expressive three image frames from 327 sequences of 118 subjects.

DTCTH achieves better performance (98.98%) with lower computational cost on CK dataset than LBP [13], boosted LBP [18], NABP [17], LGP [16], LTP [2], HOG [31], LDN [50], and CENTRIST [7] which are presented in Tables 10 and 11. DTCTH also achieves better accuracies than computationally costly Gabor features [52] (89.40%) on this dataset.

Table 12 demonstrates the confusion matrix of 6 different expressions in CK dataset. From this matrix, it can be seen that DTCTH performs better in all the basic expressions. Anger, sadness, and fear show comparatively lower performance than other expressions which is generally happened in expression recognition in CK dataset (see Table 13). However, other existing approaches provide inferior results in these expressions than DTCTH.

Table 11 presents the results on CK+ dataset which shows that DTCTH (93.99%) outperforms existing state-of-the-art approaches such as LDN, NABP, LTP, LBP, LGP, HOG, and CENTRIST. It is noteworthy to mention here that DTCTH outperforms even deep learning based methods described in [101] and [104] on both CK and CK+ datasets. Besides this, Table 14 demonstrates the confusion matrix of seven different expressions in CK+ dataset. From this matrix, it can be concluded that DTCTH achieves better accuracy in challenging expressions such as contempt, sadness, fear, and anger, though most of the existing techniques provide poor performance in these expressions.

Table 13 Confusion matrix of DTCTH in case of 7-class expression recognition on CK

	Anger	Disgust	Fear	Sadness	Happy	Neutral	Surprise
Anger	86.67	0.0	1.90	1.9	0.0	9.52	0.0
Disgust	0.77	95.38	0.0	0.0	0.0	3.85	0.0
Fear	0.56	0.0	95.0	0.0	0.56	3.89	0.0
Sadness	1.67	0.0	0.0	93.9	0.0	3.89	0.56
Happy	0.42	0.0	0.0	0.0	99.2	0.42	0.0
Neutral	2.71	0.21	1.67	0.21	0.63	94.58	0.0
Surprise	0.0	0.0	0.0	0.0	0.0	0.0	100

Table 14 Confusion matrix of DTCTH in case of 7-class expression recognition on CK+

	Anger	Contempt	Disgust	Fear	Sadness	Happy	Surprise
Anger	96.3	2.22	0.74	0.0	0.74	0.0	0.0
Contempt	9.26	87.04	0.0	0.0	0.0	0.0	0.0
Disgust	0.56	0.0	99.44	0.0	0.0	0.0	0.0
Fear	0.0	0.0	1.33	90.67	0.0	8.0	0.0
Sadness	11.9	1.19	0.0	0.0	85.7	0.0	1.19
Happy	0.0	0.0	0.0	0.97	0.0	99.03	0.0
Surprise	0.0	0.0	0.40	0.0	0.0	0.0	99.6

6 Conclusions

In this paper, a low-level feature representation technique namely discriminative ternary census transform histogram (DTCTH) is proposed where we have shown the requirements of a low-level descriptor and introduced a way to achieve those. Rigorous experiments on five different applications including nine different datasets demonstrate that DTCTH has more discrimination ability than other existing state-of-the-art low-level descriptors. Our approach outperforms other methods that include several high-level representations for different applications. This is because DTCTH has the ability to capture the prominent features that are stable in the presence of noise and different lighting conditions.

For calculating the threshold of DTCTH, we describe a way that combines Jenks' and Weber's law. We also provide a low-cost approximation that we have found empirically. Further research can be carried out on this issue to obtain a better approximation. Moreover, the incorporation of color information and high-level feature representation like sparse coding and pooling might further boost the performance of this descriptor which will be addressed in the future.

Acknowledgements
We are really grateful to the anonymous reviewers for the corrections and useful suggestions that have substantially improved the paper. Further, we would like to thank Jianxin Wu for providing his source code.

Authors' contributions
MR, SR, RR, BMMH, and MS have contributed in designing, developing, and analyzing the methodology, performing the experimentation, and writing and modifying the manuscript. All the authors have read and approved the final manuscript.

Competing interests
The authors declare that they have no competing interests.

Publisher's Note

Received: 9 March 2016 Accepted: 28 March 2017
Published online: 28 April 2017

References

1. T Ojala, M Pietikäinen, T Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern. Anal. Mach. Intell. **24**(7), 971–987 (2002)
2. X Tan, B Triggs, Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. Image Proc. **19**(6), 1635–1650 (2010)
3. M Shoyaib, M Abdullah-Al-Wadud, O Chae, A noise-aware coding scheme for texture classification. Sensors. **11**(8), 8028–8044 (2011)
4. R Zabih, J Woodfill, in *Computer Vision–ECCV'94*. Non-parametric local transforms for computing visual correspondence (Springer, Stockholm, 1994), pp. 151–158
5. S Lazebnik, C Schmid, J Ponce, in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, vol. 2 (IEEE, New York, 2006), pp. 2169–2178
6. Y Xiao, J Wu, J Yuan, mcentrist: a multi-channel feature generation mechanism for scene categorization. IEEE Trans. Image Process. **23**(2), 823–836 (2014)
7. J Wu, JM Rehg, Centrist: a visual descriptor for scene categorization. IEEE Trans. Pattern. Anal. Mach. Intell. **33**(8), 1489–1501 (2011)
8. X Qi, R Xiao, C-G Li, Y Qiao, J Guo, X Tang, Pairwise rotation invariant co-occurrence local binary pattern. IEEE Trans. Pattern. Anal. Mach. Intell. **36**(11), 2199–2213 (2014)
9. D Huang, C Zhu, Y Wang, L Chen, Hsog: a novel local image descriptor based on histograms of the second-order gradients. IEEE Trans. Image Process. **23**(11), 4680–4695 (2014)
10. A Oliva, A Torralba, Modeling the shape of the scene: a holistic representation of the spatial envelope. Int. J. Comput. Vis. **42**(3), 145–175 (2001)
11. G Griffin, A Holub, P Perona, Caltech-256 object category dataset (2007)
12. Gemert van, JC, J-M Geusebroek, CJ Veenman, AW Smeulders, in *Computer Vision–ECCV 2008*. Kernel codebooks for scene categorization (Springer, 2008), pp. 696–709
13. T Ahonen, A Hadid, M Pietikainen, Face description with local binary patterns: application to face recognition. IEEE Trans. Pattern. Anal. Mach. Intell. **28**(12), 2037–2041 (2006)
14. D Maturana, D Mery, A Soto, in *Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference On*. Learning discriminative local binary patterns for face recognition (IEEE, Santa Barbara, 2011), pp. 470–475
15. L Zhang, R Chu, S Xiang, S Liao, SZ Li, in *Advances in Biometrics*. Face detection based on multi-block lbp representation (Springer, Seoul, 2007), pp. 11–18
16. B Jun, I Choi, D Kim, Local transform features and hybridization for accurate face and human detection. IEEE Trans. Pattern. Anal. Mach. Intell. **35**(6), 1423–1436 (2013)
17. MM Rahman, S Rahman, M Kamal, EK Dey, M Abdullah-Al-Wadud, M Shoyaib, in *Computer and Information Technology (ICCIT), 2015 18th International Conference On*. Noise adaptive binary pattern for face image analysis (IEEE, Dhaka, 2015), pp. 390–395
18. C Shan, S Gong, PW McOwan, Facial expression recognition based on local binary patterns: a comprehensive study. Image Vision Comput. **27**(6), 803–816 (2009)
19. M Shoyaib, JM Youl, MM Alam, O Chae, in *Proceedings of the 2010 13th International Conference on Computer and Information Technology (ICCIT)*. Facial expression recognition based on a weighted local binary pattern (IEEE, Dhaka, 2010), pp. 321–324
20. MM Rahman, S Rahman, EK Dey, M Shoyaib, A gender recognition approach with an embedded preprocessing. Int. J. Inf. Technol. Comput. Sci. (IJITCS). **7**(7), 19 (2015)
21. S Murala, R Maheshwari, R Balasubramanian, Local tetra patterns: a new feature descriptor for content-based image retrieval. IEEE Trans. Image Process. **21**(5), 2874–2886 (2012)
22. RT Collins, A Lipton, T Kanade, H Fujiyoshi, D Duggins, Y Tsin, D Tolliver, N Enomoto, O Hasegawa, P Burt, et al, *A system for video surveillance and monitoring*, Tech. Rep. CMU-RI-TR-00-12. (Robotics Institute, Carnegie Mellon University, Pittsburgh, 2000)
23. V Pavlovic, R Sharma, TS Huang, et al, Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans. Pattern. Anal. Mach. Intell. **19**(7), 677–695 (1997)
24. A Vailaya, MA Figueiredo, AK Jain, H-J Zhang, Image classification for content-based indexing. IEEE Trans. Image Process. **10**(1), 117–130 (2001)
25. AK Jain, A Ross, S Prabhakar, An introduction to biometric recognition. IEEE Trans. Circ. Syst. Video Technol. **14**(1), 4–20 (2004)
26. AP Dhawan, *Medical image analysis*. (John Wiley & Sons; IEEE Press, 2011)
27. S Rahman, MM Rahman, K Hussain, SM Khaled, M Shoyaib, in *Computer and Information Technology (ICCIT), 2014 17th International Conference On*. Image enhancement in spatial domain: a comprehensive study (IEEE, Dhaka, 2014), pp. 368–373
28. K Hussain, S Rahman, S Khaled, M Abdullah-Al-Wadud, M Shoyaib, in *Software, Knowledge, Information Management and Applications (SKIMA), 2014 8th International Conference On*. Dark image enhancement by locally transformed histogram (IEEE, Dhaka, 2014), pp. 1–7
29. R Margolin, L Zelnik-Manor, A Tal, in *Computer Vision–ECCV 2014*. Otc: A novel local descriptor for scene classification (Springer, Zurich, 2014), pp. 377–391
30. DG Lowe, Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**(2), 91–110 (2004)
31. N Dalal, B Triggs, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference On*. Histograms of oriented gradients for human detection, vol. 1 (IEEE, San Diego, 2005), pp. 886–893
32. O Déniz, G Bueno, J Salido, F De la Torre, Face recognition using histograms of oriented gradients. Pattern Recogn. Lett. **32**(12), 1598–1603 (2011)
33. L-J Li, L Fei-Fei, in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference On*. What, where and who? Classifying events by scene and object recognition (IEEE, Rio de Janeiro, 2007), pp. 1–8
34. A Bosch, A Zisserman, X Muoz, Scene classification using a hybrid generative/discriminative approach. IEEE Trans. Pattern. Anal. Mach. Intell. **30**(4), 712–727 (2008)
35. H Bay, A Ess, T Tuytelaars, L Van Gool, Speeded-up robust features (surf). Comp. Vision Image Underst. (CVIU). **110**(3), 346–359 (2008)
36. K Mikolajczyk, C Schmid, A performance evaluation of local descriptors. IEEE Trans. Pattern. Anal. Mach. Intell. **27**(10), 1615–1630 (2005)
37. Y Ke, R Sukthankar, in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference On*. Pca-sift: a more distinctive representation for local image descriptors, vol. 2 (IEEE, Washington, 2004), pp. 506–513
38. Y Mu, S Yan, Y Liu, T Huang, B Zhou, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. Discriminative local binary patterns for human detection in personal album (IEEE, Anchorage, 2008), pp. 1–8
39. J Ren, X Jiang, J Yuan, G Wang, Optimizing lbp structure for visual recognition using binary quadratic programming. IEEE Signal Process. Lett. **21**(11), 1346–1350 (2014)
40. C Zhu, C-E Bichot, L Chen, Image region description using orthogonal combination of local binary patterns enhanced with color information. Pattern Recog. **46**(7), 1949–1963 (2013)
41. D Huang, C Shan, M Ardabilian, Y Wang, L Chen, Local binary patterns and its application to facial image analysis: a survey. IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. **41**(6), 765–781 (2011)
42. S Zahid Ishraque, M Shoyaib, M Abdullah-Al-Wadud, MM Hoque, O Chae, A local adaptive image descriptor. New Rev. Hypermedia Multimedia. **19**(3-4), 286–298 (2013)
43. O Dabeer, S Chaudhuri, Analysis of an adaptive sampler based on Weber's law. IEEE Trans. Signal Process. **59**(4), 1868–1878 (2011)
44. Z Guo, L Zhang, D Zhang, A completed modeling of local binary pattern operator for texture classification. IEEE Trans. Image Process. **19**(6), 1657–1663 (2010)
45. E Tola, V Lepetit, P Fua, Daisy: an efficient dense descriptor applied to wide-baseline stereo. IEEE Trans. Pattern. Anal. Mach. Intell. **32**(5), 815–830 (2010)

46. M Brown, G Hua, S Winder, Discriminative learning of local image descriptors. IEEE Trans. Pattern. Anal. Mach. Intell. **33**(1), 43–57 (2011)
47. L Robertson, Methods and innovations for multimedia database content management/current trends and future practices for digital literacy and competence. Aust. Libr. J. **62**(2), 170–171 (2013)
48. Y Ma, P Cisar, in *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference On*. Event detection using local binary pattern based dynamic textures (IEEE, Miami, 2009), pp. 38–44
49. M Shoyaib, M Abdullah-Al-Wadud, SZ Ishraque, O Chae, in *Belief Functions: Theory and Applications*. Facial expression classification based on dempster-shafer theory of evidence (Springer, Compiègne, 2012), pp. 213–220
50. A Ramirez Rivera, J Rojas Castillo, O Chae, Local directional number pattern for face analysis: face and expression recognition. IEEE Trans. Image Process. **22**(5), 1740–1752 (2013)
51. ul Hussain, S, B Triggs, in *Computer Vision–ECCV 2012*. Visual recognition using local quantized patterns (Springer, Florence, 2012), pp. 716–729
52. MS Bartlett, G Littlewort, I Fasel, JR Movellan, in *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference On*. Real time face detection and facial expression recognition: development and applications to human computer interaction, vol. 5 (IEEE, 2003), pp. 53–53
53. W Gu, C Xiang, Y Venkatesh, D Huang, H Lin, Facial expression recognition using radial encoding of local gabor features and classifier synthesis. Pattern Recognit. **45**(1), 80–91 (2012)
54. D Weng, Y Wang, M Gong, D Tao, H Wei, D Huang, Derf: distinctive efficient robust features from the biological modeling of the p ganglion cells. IEEE Trans. Image Process. **24**(8), 2287–2302 (2015)
55. L-J Li, H Su, L Fei-Fei, EP Xing, in *Advances in Neural Information Processing Systems*. Object bank: a high-level image representation for scene classification & semantic feature sparsification (NIPS, Hyatt Regency, Vancouver, 2010), pp. 1378–1386
56. M Pandey, S Lazebnik, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. Scene recognition and weakly supervised object localization with deformable part-based models (IEEE, Barcelona, 2011), pp. 1307–1314
57. Y Yang, S Newsam, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. Spatial pyramid co-occurrence for image classification (IEEE, Barcelona, 2011), pp. 1465–1472
58. Z Wang, Y Hu, L-T Chia, in *Computer Vision–ECCV 2010*. Image-to-class distance metric learning for image classification (Springer, Heraklion, 2010), pp. 706–719
59. O Boiman, E Shechtman, M Irani, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. In defense of nearest-neighbor based image classification (IEEE, Anchorage, 2008), pp. 1–8
60. J Yang, K Yu, Y Gong, T Huang, in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference On*. Linear spatial pyramid matching using sparse coding for image classification (IEEE, Miami, 2009), pp. 1794–1801
61. A Shabou, H LeBorgne, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On*. Locality-constrained and spatially regularized coding for scene categorization (IEEE, RI, USA, 2012), pp. 3618–3625
62. J Wang, J Yang, K Yu, F Lv, T Huang, Y Gong, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference On*. Locality-constrained linear coding for image classification (IEEE, San Francisco, 2010), pp. 3360–3367
63. GL Oliveira, ER Nascimento, AW Vieira, MF Campos, in *Robotics and Automation (ICRA), 2012 IEEE International Conference On*. Sparse spatial coding: a novel approach for efficient and accurate object recognition (IEEE, St. Paul, 2012), pp. 2592–2598
64. Z Wang, J Feng, S Yan, H Xi, Linear distance coding for image classification. IEEE Trans. Image Process. **22**(2), 537–548 (2013)
65. S Gao, IW-H Tsang, L-T Chia, in *Computer Vision–ECCV 2010*. Kernel sparse representation for image classification and face recognition (Springer, Heraklion, 2010), pp. 1–14
66. S Gao, IW-H Tsang, L-T Chia, Laplacian sparse coding, hypergraph laplacian sparse coding, and applications. IEEE Trans. Pattern. Anal. Mach. Intell. **35**(1), 92–104 (2013)
67. L Bo, C Sminchisescu, in *Advances in Neural Information Processing Systems*. Efficient match kernel between sets of features for visual recognition (NIPS, Vancouver, 2009), pp. 135–143
68. L Liu, L Wang, X Liu, in *Computer Vision (ICCV), 2011 IEEE International Conference On*. In defense of soft-assignment coding (IEEE, Barcelona, 2011), pp. 2486–2493
69. A Bosch, A Zisserman, X Muñoz, in *Computer Vision–ECCV 2006*. Scene classification via plsa (Springer, Graz, 2006), pp. 517–530
70. A Krizhevsky, I Sutskever, GE Hinton, in *Advances in Neural Information Processing Systems*. Imagenet classification with deep convolutional neural networks (NIPS, Lake Tahoe, 2012), pp. 1106–1114
71. B Zhou, A Lapedriza, J Xiao, A Torralba, A Oliva, in *Advances in Neural Information Processing Systems*. Learning deep features for scene recognition using places database (NIPS, Montreal, 2014), pp. 487–495
72. K Chatfield, K Simonyan, A Vedaldi, A Zisserman, Return of the devil in the details: delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
73. L Fei-Fei, R Fergus, P Perona, Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. Comp. Vision Image Underst. (CVIU). **106**(1), 59–70 (2007)
74. L Yang, R Jin, R Sukthankar, F Jurie, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. Unifying discriminative visual codebook generation with classifier training for object category recognition (IEEE, Anchorage, 2008), pp. 1–8
75. S Maji, AC Berg, J Malik, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. Classification using intersection kernel support vector machines is efficient (IEEE, Anchorage, 2008), pp. 1–8
76. P Dollar, C Wojek, B Schiele, P Perona, Pedestrian detection: an evaluation of the state of the art. IEEE Trans. Pattern. Anal. Mach. Intell. **34**(4), 743–761 (2012)
77. H Zhang, AC Berg, M Maire, J Malik, in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference On, vol. 2*. Svm-knn: discriminative nearest neighbor classification for visual category recognition (IEEE, New York, 2006), pp. 2126–2136
78. JC Platt, N Cristianini, J Shawe-Taylor, in *Advances in Neural Information Processing Systems*. Large margin dags for multiclass classification, vol. 12 (MIT Press, Denver, 1999), pp. 547–553
79. J Wu, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference On*. Power mean svm for large scale visual classification (IEEE, RI, USA, 2012), pp. 2344–2351
80. University of Kansas. Department of Geography, G Jenks, *Optimal data classification for choropleth maps*, (1977)
81. RG Cromley, A comparison of optimal classification strategies for choroplethic displays of spatially aggregated data. Int. J. Geogr. Inf. Syst. **10**(4), 405–424 (1996)
82. T Kanade, JF Cohn, Y Tian, in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference On*. Comprehensive database for facial expression analysis (IEEE, Grenoble, 2000), pp. 46–53
83. R-E Fan, K-W Chang, C-J Hsieh, X-R Wang, C-J Lin, Liblinear: a library for large linear classification. J. Mach. Learn. Res. **9**, 1871–1874 (2008)
84. P Jain, B Kulis, K Grauman, in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference On*. Fast image search for learned metrics (IEEE, Anchorage, 2008), pp. 1–8
85. G Serra, C Grana, M Manfredi, R Cucchiara, Gold: Gaussians of local descriptors for image representation. Comp. Vision Image Underst. (CVIU). **134**, 22–32 (2015)
86. J Chen, Q Li, Q Peng, KH Wong, Csift based locality-constrained linear coding for image classification. Pattern. Anal. Applic. **18**(2), 441–450 (2015)
87. M Heikkilä, M Pietikäinen, C Schmid, Description of interest regions with local binary patterns. Pattern Recognit. **42**(3), 425–436 (2009)
88. A Borji, L Itti, in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference On*. Human vs. computer in scene and object recognition (IEEE, Columbus, 2014), pp. 113–120
89. L Xie, J Wang, W Lin, B Zhang, Q Tian, in *Proceedings of the IEEE International Conference on Computer Vision*. Ride: reversal invariant descriptor enhancement (IEEE, Santiago, 2015), pp. 100–108

90. F Perronnin, J Sánchez, T Mensink, in *European Conference on Computer Vision*. Improving the fisher kernel for large-scale image classification (Springer, Heraklion, 2010), pp. 143–156
91. A Quattoni, A Torralba, in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference On*. Recognizing indoor scenes (IEEE, Miami, 2009), pp. 413–420
92. A Sharif Razavian, H Azizpour, J Sullivan, S Carlsson, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Cnn features off-the-shelf: an astounding baseline for recognition (IEEE, Columbus, 2014), pp. 512–519
93. Z Zuo, G Wang, B Shuai, L Zhao, Q Yang, X Jiang, in *Computer Vision–ECCV 2014*. Learning discriminative and shareable features for scene classification (Springer, Zurich, 2014), pp. 552–568
94. J Zhu, L-J Li, L Fei-Fei, EP Xing, in *Advances in Neural Information Processing Systems*. Large margin learning of upstream scene understanding models (Curran Associates, Inc., Vancouver, 2010), pp. 2586–2594
95. B Lei, E-L Tan, S Chen, D Ni, T Wang, Saliency-driven image classification method based on histogram mining and image score. Pattern Recog. **48**(8), 2567–2580 (2015)
96. Y-L Boureau, F Bach, Y LeCun, J Ponce, in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference On*. Learning mid-level features for recognition (IEEE, San Francisco, 2010), pp. 2559–2566
97. O Söderkvist, Computer vision classification of leaves from swedish trees. Master's Thesis, Linkoping University (2001)
98. H Ling, DW Jacobs, Shape classification using the inner-distance. IEEE Trans. Pattern. Anal. Mach. Intell. **29**(2), 286–299 (2007)
99. PF Felzenszwalb, JD Schwartz, in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference On*. Hierarchical matching of deformable shapes (IEEE, Minneapolis, 2007), pp. 1–8
100. S Zhang, Y Lei, T Dong, X-P Zhang, Label propagation based supervised locality projection analysis for plant leaf classification. Pattern Recognit. **46**(7), 1891–1897 (2013)
101. J Susskind, V Mnih, G Hinton, et al, in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference On*. On deep generative models with applications to recognition (IEEE, Colorado Springs, 2011), pp. 2857–2864
102. I Cohen, N Sebe, A Garg, LS Chen, TS Huang, Facial expression recognition from video sequences: temporal and static modeling. Comp. Vision Image Underst. (CVIU). **91**(1), 160–187 (2003)
103. P Lucey, JF Cohn, T Kanade, J Saragih, Z Ambadar, I Matthews, in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference On*. The extended Cohn-Kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression (IEEE, San Francisco, 2010), pp. 94–101
104. M Liu, S Li, S Shan, X Chen, in *Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops On*. Au-aware deep networks for facial expression recognition (IEEE, Shanghai, 2013), pp. 1–6