

---

# Effective Compression of Center Symmetric Local Binary Pattern

JUNAID BABER\*, MAHEEN BAKHTYAR\*, IHSAN ULLAH\*, MARIAM REHMAN\*\*, AND MUHAMMAD KHALID\*

RECEIVED ON 11.11.2015 ACCEPTED ON 16.02.2016

## ABSTRACT

In this paper, we propose simple and effective compression of CSLBP (Center Symmetric Local Binary Pattern) descriptors, which is a textured based operator and mostly used as key point descriptor. With default parameters for computation, it is 256-length descriptor for each keypoint or affine patch. CSLBP is an extended form of LBP (Local Binary Patterns). The calculation of CSLBP descriptor is effective, robust, and straightforward for different image transformations for instance; image blurring and illumination alteration. However, an improvement in time and space consumption of CSLBP can be attained by means of simple compression. For this reason, CSLBP is a smart choice for smart phones as well as large databases. We reduce the descriptor length (dimensions) upto 50% without applying any techniques of dimensionality reduction like PCA (Principle Component Analysis) or LDA (Linear Discriminant Analysis). The compressed CSLBP descriptor is denoted as C-CSLBP. The performance of C-CSLBP is evaluated on state-of-the-art datasets using standard metrics. It is quantitatively shown by experiments that C-CSLBP is equivalently effective compared to CSLBP despite of reduced dimensions.

**Key Words:** Dimensionality Reduction, Center Symmetric Local Binary Pattern, Q-CSLBP, Image Retrieval, Descriptor.

## 1. INTRODUCTION

One of the major issues of image processing and computer vision research is to find out images which are similar in part or whole to a query image. This has several important applications such as video indexing, image retrieval, image classification, texture recognition, and object recognition. A large number of approaches have been proposed which mainly work in three phases: (1) to determine keypoint locations that are repeatable; (2) estimate a descriptor vector that evaluate the local image patch properties around every keypoint; (3) then a similarity measure based on keypoint

descriptors pairs is employed to match keypoints. SIFT (Scale Invariant Feature Transform) [1], SURF (Speeded UP Robust Features) [2], and ORB (Oriented FAST and Rotated BRIEF) [3] are a few of the keypoint detection and description algorithms that have achieved substantial success.

Many consider the SIFT descriptor as benchmark standard for keypoint descriptor. Its resilience against image transformations [4,5] has made it favorable for a number of applications including image classification [6], object recognition [1], image stitching [7], and image/video copy

---

\* Department of Computer Science & Information Technology, University of Balochistan, Quetta.

\*\* Department of Computer Science, Lahore College for Women University, Lahore.

detection [8]. The SIFT descriptor is the representation of gradient orientation histograms. To estimate SIFT descriptor, the patch surrounding every keypoint is selected and divided into 4x4 grid. For all cells, the gradient magnitudes and orientations are estimated for all pixels. The gradient orientations are quantized into 8 directions and the histograms of quantized orientations are computed. During the histograms computation, each sample added to histograms are weighted by their gradient magnitude and Gaussian weight. The main problem with SIFT is computational cost and complexity. Marko, et. al. [9] proposed texture based descriptor known as CSLBP. In this research, it has been argued that CSLBP is computationally fast and simple to compute. It is also shown that CSLBP descriptor has similar performance compared to SIFT under different transformations. CSLBP is 256-D descriptor based on its default parameters.

The use of keypoint descriptor matching is confronted by two major issues for image search problems in making an effort to attack problems with particularly large domains for instance; detection of a copy of video on the Web. The origin of both of these problems is the elevated dimensionality of common keypoint descriptors including SIFT and CSLBP. The primary issue is concerned with storage space. Thousands of keypoint are possible for just one image, and the set of raw descriptors might require more storage space than image. Indexing the frames of a collection of videos by, say, a movie studio, in order to perform illegal copy detection is not trivial. The second issue is the amount of time needed in similarity computation of two descriptors. To evaluate similarity involves complex measures like Euclidean distance which is  $O(n)$ , where  $n$  represents the descriptor vector dimensionality. Thus, to search for an image from millions of images that is similar to a query image has enormous cost in terms of money and time. In this way, the time needed to compute the distance of descriptor can considerably lessen the cost of image search at large scale.

The descriptor matching speed can be enhanced in number of possible ways. Firstly, dimensionality reduction mechanisms, like PCA (Principal Component Analysis) can be applied to lessen dimensionality. A SIFT descriptor based on PCA called (PCA-SIFT) is proposed and evaluated in [10]. The lower dimensionality makes this method faster to match. Moreover, it is useful for task of image retrieval. Nevertheless, a discriminative illustration intended for recognition is not essentially provided by PCA, since it produces global basis vectors and is unsupervised [11]. Secondly, "smart" lower-dimensional descriptors can possibly be used to drop out information that is potentially not very effective for robust matching and to retain the information which is not influenced as a result of image transformations. The SURF descriptor [2] version based on 64-element is one such example. The exploitation of binary descriptors for instance; BRIEF (Binary Robust Independent Elementary Features) [12], BRISK (Binary Robust Invariant Scalable Keypoints) [13], ORB [3], and CARD (Compact And Real-time Descriptors) [14] is a third option. The advantage with binary descriptors is the requirement of a small storage space, thus huge amount of descriptors can be transferred through the storage hierarchy in a reduced amount of time. Matching and search operations are accelerated through better storage throughput. However, a major limitation of binary descriptors is their tendency to provide poor performance under image deformations like alterations in scale and viewpoint. In a fourth category of techniques, the focus is on finding estimated closest neighbors founded on hash codes of a query descriptor. A hash function is employed for coding and quantization of every descriptor that could be either learned from data [15-18] or randomized. These techniques also produce binary descriptors, although data is employed to learn discriminative projections, these techniques are able to reduce memory as well as computational cost to a great extent. Approaches of a fifth category are based on vector

quantization techniques. In these approaches, a large collection of features vectors is made, and after that a clustering algorithm is employed that map every vector of a feature space to a distinct group. The BoVW (Bag of Visual Words) model is one of the common approaches which is based on quantization. In BoVW, a sparse frequency vector is used to represent an image. The sparse frequency vector is based on a group of cluster IDs of feature descriptor. This technique is applied in various applications such as image retrieval, classification, and image copy detection [19]. Nevertheless, hashing quantization as well as vector quantization has their limitations such that both are not naturally adaptive. Hashcode or codebook size impact their performance. Moreover, valuable information provided by raw descriptors for discrimination may perhaps discarded by using quantization that may be important regarding discrimination [20,21]. In addition, the codebook or Hashcode that is learned from one set of images will not be generalized to a new set of images.

Lately, some techniques have been proposed that make use of vector quantization. They introduce deterministic mappings that are simple and do not have need of datasets for training. Scalar quantization [22,23] is one such alternative which allows to provide an equivalence relation that is fixed and based on the feature space as well as code every equivalence class by means of a bit vector.

In this paper, a new method C-CSLBP is proposed that is based on CSLBP. C-CSLBP reduces high dimensional CSLBP into lower dimensional vector. We reduce the memory requirement of CSLBP 256 bytes (1024 bytes in case of 32-bit floating point numbers utilization) to just 128 bytes. The fixed compression is employed; it does not involve offline learning of visual words which saves the time for training in addition to the problem of adaptiveness to other new features.

The proposed compressed descriptor, C-CSLBP which is an extension of our previous work [24], is thus valuable as it decreases the computational and storage resources needed for large scale applications that provide image to image matching. In this paper we extend the experimental setup and investigate the descriptor performance for large scale image retrieval. However, our core methodology for descriptor compression remains the same. Furthermore, we look forward to its convenience on mobile gadgets that have inadequate resources in terms of storage and computation for instance; tablets and smartphones.

Extensive experimentation is accomplished to validate the proposed technique as well as to evaluate the performance of the proposed approach. Based on the experiments, our observation is that C-CSLBP performs equally to discriminate and robustly compare against the existing state of the art descriptors that are designed for feature matching under various geometric along with photometric alterations. Moreover, it has similar performance in image retrieval for large collections.

## 2. RELATED WORK

The state of the art feature point descriptors can be categorized into two subcategories: one category is based on gradient histograms [1,2,4,10,14] and the other is based on local pixel intensity differences lacking gradient computations [3,9,12,25]. The latter category of descriptors is termed as intensity based descriptors.

LBP [26] are amongst the commonly employed features that are based on intensity, and are applied in human detection [27], face recognition [28], background subtraction [25] and 3D (Three Dimensional) textured surface recognition [29]. Computational simplicity makes the LBP operator significantly attractive. To estimate LBP features of an image patch, a pixels gray level intensity is compared with neighbor pixels  $N$  having distance  $r$  from

pixel. The values recommended for  $N$  and  $r$  are  $N = 8$  and  $r = 1$ , respectively. The consequence of comparisons provides a binary vector that express the correlation between the pixels gray level intensity to all of pixels neighbors. After that, based on these binary vectors, a histogram is computed over the affine region of keypoint. The number of entries in histogram are  $2^N$  because the possible values of local binary pattern operator are  $2^N$ . Certainly, the huge number of bins is the central drawback of LBP. For this reason, LBP is not very convenient in support of local image patches. However, it is valuable when considering complete representation of images that is used for image classification.

An adaptation of LBP known as CSLBP [9] is proposed which is a quantized form of LBP. In CSLBP, instead of comparing intensity of every pixel to its  $N$  neighbors, just center symmetric pairs are considered during comparison [9]. This consequence in reduced size of histogram for CSLBP; for instance, size of histogram of  $CSLBP_{r,N}$  is  $2^{N/2}$ . For this reason, CSLBP is additional suitable when considering local keypoint description as compared to LBP itself. However,  $CSLBP_{r,N}$  is efficient and computationally straightforward, it has a reduced amount of tolerance to rotation as well as scale variations as compared to SIFT, and the dimensionality of CSLBP descriptor is still naturally more as compared to SIFT descriptor.

Calonder et. al. [12] propose BRIEF which is a binary descriptor. Likewise, LBP and CSLBP, BRIEF is estimated for a given patch by applying a set of intensity based binary tests. The central difference between descriptors of BRIEF and LBP is the comparisons pattern of pixel which is employed to all patches to acquire the bit descriptor vector. This pattern is selected randomly in advance instead of employing a particular pattern like center symmetric pairs. The authors discover that Gaussian distribution works fine for test pixels selection

[12,3]. BRIEF is computationally fast as it employs simple pairwise comparisons that result in a bit string. Under various transformations like illumination and blur, the robustness of BRIEF is alike robustness of SIFT, although it is pretty sensitive to scale variations and rotation. For this reason, BRIEF is not suitable for applications that require robustness to scale variations and rotation.

Ruble et. al. [3] propose an adaptation of BRIEF namely ORB. The ORB descriptor is noise resistant as well as rotation invariant. Similar to BRIEF, ORB make use of FAST features for detection of keypoints [30]. FAST is neither scale resistant nor rotation invariant, on the other hand it is extremely speedy to calculate. The ORB estimates the centroid in addition to pixel values moments of image patch. Then, ORB exploits these moments to acquire the point of reference for all features to formulate the approach rotation invariant.

Leutenegger et. al. [13] propose a descriptor which consider both scale resistant and rotation invariant that is BRISK descriptor. A binary string is estimated in BRISK similar to ORB and BRIEF. For comparisons of pixels intensity, symmetric patterns are captured in BRISK like DAISY descriptor [31] (despite the fact that BRISK is considered to work on keypoints whereas DAISY is designed to work on dense points). Computation of BRISK descriptor is more expensive. Furthermore, BRISK requires additional space as compared to ORB and BRIEF [5].

Even though keypoint descriptors that are based on intensity are easy to calculate and have proven to be greatly successful in object detection as well as image classification, keypoint descriptors based on gradient perform better as compared to intensity based descriptors on image retrieval as well as object detection tasks. Now we provide review on keypoint descriptors that are based on gradient.

SIFT [1] is the most extensively employed keypoint detection approach based on gradient. SIFT is applied in image retrieval, objection detection, image classification, image copy detection and robotics. To calculate a descriptor based on SIFT, the region surrounding a keypoint is selected and divided into a grid of 4x4 (in almost every implementation). For every cell, the orientation  $\theta(x,y)$  as well as gradient magnitude  $g(x,y)$  is estimated for every pixel. After that every gradient orientation is quantized into any direction of total of eight directions. After that based on quantized gradient orientations, a weighted histogram is calculated. In every pixels histogram, the weight is the pixel's gradient magnitude that is transformed by means of a keypoint centered circular Gaussian function. The scale of circular Gaussian function is 1.5 times the scale of keypoint. The modulation based on Gaussian function signifies that the contribution of pixels which are close to the keypoint is more powerful as compared to the pixels faraway. At last, all cells gradient orientation histograms are combined into a single vector.

Bay et. al. [2] propose a SIFT adaptation that is SURF. SIFT is computationally slower as compared to SURF. SURF is faster generally as a result of extensive exploit of Haar like filter responses that are estimated rapidly by means of integral images. In a type of SURF, the approach further optimizes through dividing the SIFT descriptor dimensionality partially. There are a variety of applications that employ SURF effectively. These applications include image recognition, feature tracking [32], shot boundary detection [33,34] and video scene detection [34,35]. On the other hand, Rublee, et. al. [3] illustrate that the performance of SIFT is better as compared to SURF regarding feature matching that is based on rotation based transformations.

GLOH [4] is one more famous approach that is based on gradient. In this approach, descriptors are estimated based on a log polar grid. Three circles are positioned around the keypoint. These circles are concentric and each having different radius (6, 11, and 15). Based on angular directions, the two external circles are divided into eight, whereas the third inner circle is left undivided. In this way, 17 local areas for a keypoint are obtained. Quantization is applied on gradient orientations and a histogram of 16 bins is estimated for all regions. The consequential descriptor is a vector of 272 elements containing 17 histograms of gradient orientation of 17 regions. In view of the fact that the raw descriptor based on GLOH is pretty lengthy, this approach lessens the dimensionality of descriptor to 128 by means of PCA that is estimated based on 47,000 training image patches.

Ambai et. al. [14] propose one more method that is based on gradient orientation histograms. The method is known as CARD. In CARD, lookup tables are employed to accelerate the gradient orientation histograms computation, and consequential descriptor is binarized onto a lower dimensional subspace by means of the projection signs of descriptor vector. As a consequence, CARD is noticeably faster as compared to SIFT (16 times faster revealed by experimentation) and exploits a reduced amount of storage space. On the other hand, the robustness of SIFT to image transformations is better as compared to that of CARD.

Linear dimensionality reduction methods are employed by both CARD and GLOH. Ke et. al. [10] introduced the general scheme of dimensionality reduction intended for keypoint descriptors. In their algorithm PCASIFT, PCA is employed to decrease the dimensions of SIFT descriptor up to 36 dimensions. Some other authors have revealed in their research that, to reduce dimensionality, nonlinear approaches perform well as compared to PCA-SIFT in several cases [11].

In recent times, several approaches have emerged that are based on binary codes learning for on hand descriptors. This is achieved through projecting existing descriptors onto discriminative hyper planes. The examples include supervised hashing with kernels [36], Semi-supervised hashing [17], LSH [15,16], SH (Spectral Hashing) [37], D-BRIEF [38], and LDA Hash [18]. Weiss, et. al. [37] propose an approach that employ compact binary codes for descriptors encoding. In this approach, problem of hashing is devised as a problem of graph partitioning. Strecha et. al. [18] applies LDA to descriptors acquired by SIFT. This approach employs discriminative projections to acquire binary codes. Trzcinski et. al. [38] propose Discriminative BRIEF (D-BRIEF) that make use of discriminative projections approach on keypoint patches. To acquire binary descriptors, this approach threshold the coordinates of keypoint patches. D-BRIEF needs just 4 bytes. In spite of especially reduced space (just  $2^{32}$  distinct features), the performance of D-BRIEF is comparable to performance of SIFT. Our experiments revealed that the performance of D-BRIEF slowly reduces when distortion becomes further severe.

SIFT descriptors are quantized prior to indexing and matching for large-scale applications. Sivic and Zisserman [39] propose a model intended for SIFT descriptors known as bag of visual word model. They reveal approach's effectiveness on video matching along with indexing. In this approach, descriptors are quantized to cluster centroids that are acquired as a result of k-means algorithm, and then the image is characterized through a visual words based sparse frequency vector. Scalability is the central drawback of this technique, in view of the fact that computation of k-means algorithm is expensive particularly for large values of k. Nister et. al. [40] introduce a vocabulary tree to facilitate hierarchical quantization of descriptors to extend visual word model. The authors illustrate that performance can be improved as compared

to flat k-means by employing large vocabulary. When considering large scale object and image retrieval applications, the bag of visual words model turned out to be the standard framework. Several adaptations of this approach have been presented [19,22,41,42]. One extended model is that of J'egou et. al. [41]. In this model, SIFT descriptor is divided into various parts, then every part is quantized by means of separate k-means models, and then as a final point centroids are concatenated. Zhou et. al. [22] propose a quantization of a SIFT vector. This approach employs median value in the vector for thresholding all elements of the descriptor vector. The resultant descriptor consists of 128 bits. This approach is named as SQ (Scalar Quantization). SQ provides a vector of 256-bit by means of two thresholds in order to reduce quantization error.

### 3. CSLBP COMPRESSION

As explained above, CSLBP is an extension of LBP. We briefly explain LBP and CSLBP in this section to make paper self-contained.

In LBP, each pixel value  $p$  is compared with its  $N$  neighbors with radial distance  $r$ . For every pixel  $p$ , there are  $N$  comparisons and the output for each.

pixel  $p$  is of  $N$  bits and that can be represented by decimal number:

$$LBP_{N,r}(p) = \sum_{i=1}^N s(|n_i| - |p|) 2^{i-1}$$

$$s(j) = \begin{cases} 1 & j \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

where  $|p|$  is the pixel value and  $n_i$  are the neighbor pixels at the radial distance  $r$ . In practice, the values of  $N$  and  $r$  are 8 and 1, respectively. For given image or patch, the

histogram of LBP is estimated where the length of histogram is  $2^N$ . The length of LBP histogram can be reduced by uniform patterns. The LBP histogram is quite long and cannot be efficiently used for region descriptor [9].

The CSLBP operator is quantized representation of LBP operator as described in Equation (1). In CSLBP, instead of comparing each neighbor with  $p$ , only center-symmetric neighbors are compared as shown in Fig. 1.

$$CSLBP_{N,r,T}(p) = \sum_{i=1}^{\frac{N}{2}} s \left( |n_i| - \left| n_i + \frac{N}{2} \right| \right) 2^{i-1}$$

$$s(j) = \begin{cases} 1 & j \geq 0 \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

For given image or patch, the histogram of CSLBP is computed, the length of histogram is  $2^{N/2}$  which is quite short as compared to LBP. The suggested values for  $N, r$  and  $T$  are 8, 1, 0.01, respectively.

To compute the CSLBP descriptor, the given patch  $P$  is divided into spatial grid of  $G_x \times G_y$ , and the histogram of CSLBP is computed in each cell. Finally, all histograms

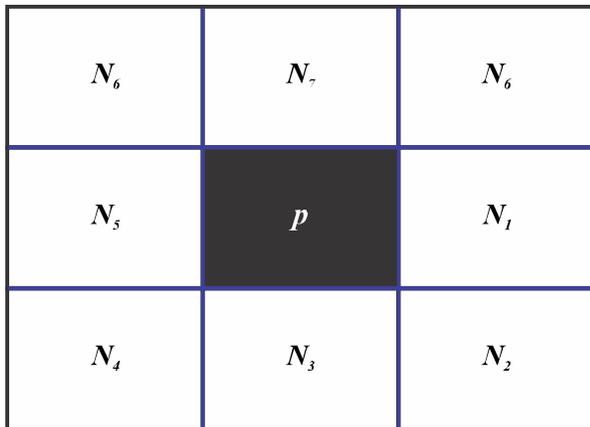


FIG. 1. EXAMPLE OF KEYPOINT  $P$  WITH ITS  $N = 8$  NEIGHBOR PIXELS AT THE RADIAL DISTANCE  $R = 1$ . IN CASE OF LBP, EACH NEIGHBOR PIXEL IS COMPARED WITH  $P$ , WHEREAS, IN CSLBP COMPARISON PAIRS ARE AS FOLLOW:  $(N_1, N_5)$ ,  $(N_2, N_6)$ ,  $(N_3, N_7)$ ,  $(N_4, N_8)$

are combined to form the descriptor. The length of CSLBP descriptor is  $G_x \times G_y \times 2^{N/2}$  which is quite often the double of SIFT descriptor.

For our experiments, the values for  $CSLBP_{N,r,T}$  are  $CSLBP_{8,1,0.01}$ , and optimal results are obtained keeping  $G_x = 4$  and  $G_y = 4$  which makes CSLBP the length of 256. CSLBP histograms from all the cells are concatenated to acquire a vector. After that, vector is normalized to unit length. After normalization, the values having long peaks are again thresholded by 0.2 to ensure that no value is greater than 0.2, as suggested by [9]. After thresholding, the vector is renormalized to unit length.

In CSLBP, for each pixel there are four comparisons between eight neighbors as shown in Fig. 1. Based on eight neighbor texture, there are 16 unique patterns using Equation (2), these patterns are represented by decimal numbers from 0-15, and CSLBP histogram of these 16 patterns for all pixels are computed for given patch  $P$ . One possible way to lessen the dimensions is to employ less number of bins instead 16. Analysis on co-occurrence distribution [43] of CSLBP patterns on VGG dataset (dataset explained in Experimental section) shows that the adjacent bins have high correlation. Therefore, information loss will be significant if dimensions are reduced by taking less number of bins.

The co-occurrence distribution also show that the symmetric patterns have less co-relation in image texture. Experiment shows that merging of symmetric patterns is more effective, symmetric patterns can be obtained by flipping the comparison operator used in Equation (2), i.e.,  $j > T$ . We examined many combinations to reduce bins by merging different patterns. However, maximum efficiency is achieved when symmetric patterns are merged, as shown in Fig. 2. We simply merge two patterns obtained by flipping the comparison operator used in Equation (2). For example, given two points  $p_1, p_2$  in region  $P$ , we get following CSLBP:

The above example shows that two different patterns of same texture are obtained by flipping the comparison operator. In framework,  $p_1$  and  $p_2$  are treated as single pattern and their decimals are put into single bin. By merging the symmetric patterns and computing the histogram,  $2^3$  length histogram is obtained. Whereas, CSLBP is  $2^4$  length histogram. The compression of CSLBP descriptor is 50% without significant loss in discrimination power. The compressed CSLBP descriptor is called C-CSLBP.

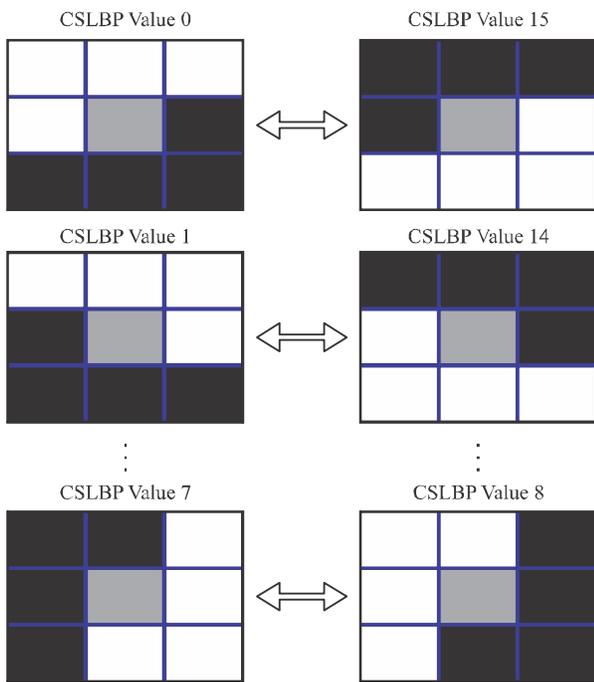


FIG. 2. C-CSLBP: MERGING THE SYMMETRIC PATTERNS

#### 4. EXPERIMENTAL EVALUATION

This section summarizes the experimental verification of CSLBP compression. The robustness of C-CSLBP under different types of image transformations with different level of degradations is evaluated.

Two datasets are employed to confirm the performance of compression. The first dataset provided by VGG (Visual Geometry Group) is used, this is standard dataset and used by many researchers [4,9,14,23,24]. The dataset is available on-line as well<sup>1</sup>. The dataset contains several kinds of photometric and geometric transformations on various type of scenes, it contains illumination change (leuven), image blur (bikes), JPEG compression (ubc), zoom and rotate (bark and boats), and viewpoint change (graffiti) as shown in Fig. 3. For each transformation, original image  $Q$  is provided with 5 gradual deformed images  $\{L_1, L_2, \dots, L_5\}$ , image  $L_i$  has severe transformation compared to  $L_{i-1}$ . Therefore, performance of descriptors matching start decreasing as transformation get more severe. All the images in test dataset are related by a homography and provided by their fundamental matrix.

IR is the second dataset that is presented by Zhou et al. [42]. IR dataset consist of 1K images that represent 36 various scenes. Challenging transformations that are usually present in situations of copy detection are applied on all the images to acquire strictly distorted images. The types of transformation used in this dataset are listed in Table 1. Fig. 4(a) illustrates an image chosen randomly from every scene, and Fig. 4(b) illustrate ten images that are selected arbitrarily from various scenes.

Example	Equation (2) with $j > T$	Flipped Comparison $j < T$									
<table border="1"> <tr><td>15</td><td>255</td><td>150</td></tr> <tr><td>75</td><td><math>p_1</math></td><td>50</td></tr> <tr><td>100</td><td>100</td><td>5</td></tr> </table>	15	255	150	75	$p_1$	50	100	100	5	<p>CSLBP (<math>p_1</math>) = 0</p>	<p>CSLBP (<math>p_1</math>) = 15</p>
15	255	150									
75	$p_1$	50									
100	100	5									
<table border="1"> <tr><td>50</td><td>25</td><td>120</td></tr> <tr><td>100</td><td><math>p_2</math></td><td>150</td></tr> <tr><td>190</td><td>95</td><td>250</td></tr> </table>	50	25	120	100	$p_2$	150	190	95	250	<p>CSLBP (<math>p_2</math>) = 15</p>	<p>CSLBP (<math>p_2</math>) = 0</p>
50	25	120									
100	$p_2$	150									
190	95	250									

We used HA (Harris Affine) keypoints [45], these points have corner like structure with low localization error and higher repeatability compared to DoG (Difference of Gaussian) keypoints which is used by SIFT algorithm [45]. For HA keypoints, multi-scale harris detector is used, each point is represented by elliptical region, computed by second moment matrix [45]. The elliptical region is mapped to circular region; circular region is rotated in the direction of dominant gradient to make it rotation invariant [1]. For descriptor computation, the circular region is normalized to 41x41 pixels in Cartesian grid, and further divided into 4x4 cells [4,9]. From each cell, histograms of CSLBP and C-CSLBP are computed and merged to their single respective vectors. The descriptor length for CSLBP is 256 and C-CSLBP is 128 only.

TABLE 1. LIST OF CHALLENGES FOR COPY DETECTION

No.	Description
T1	Camcorder image
T2	Picture in picture
T3	Insertions of pattern
T4	JPEG compression
T5	Change of illumination
T6	Cropping
T7	Blurring
T8	Image flipping
T9	Text insertion
T10	Zoom and rotate
T11	Viewpoint change
T12	Decrease in image quality - This includes combination of all eleven challenges along with image shifting, contrast, and image morphing

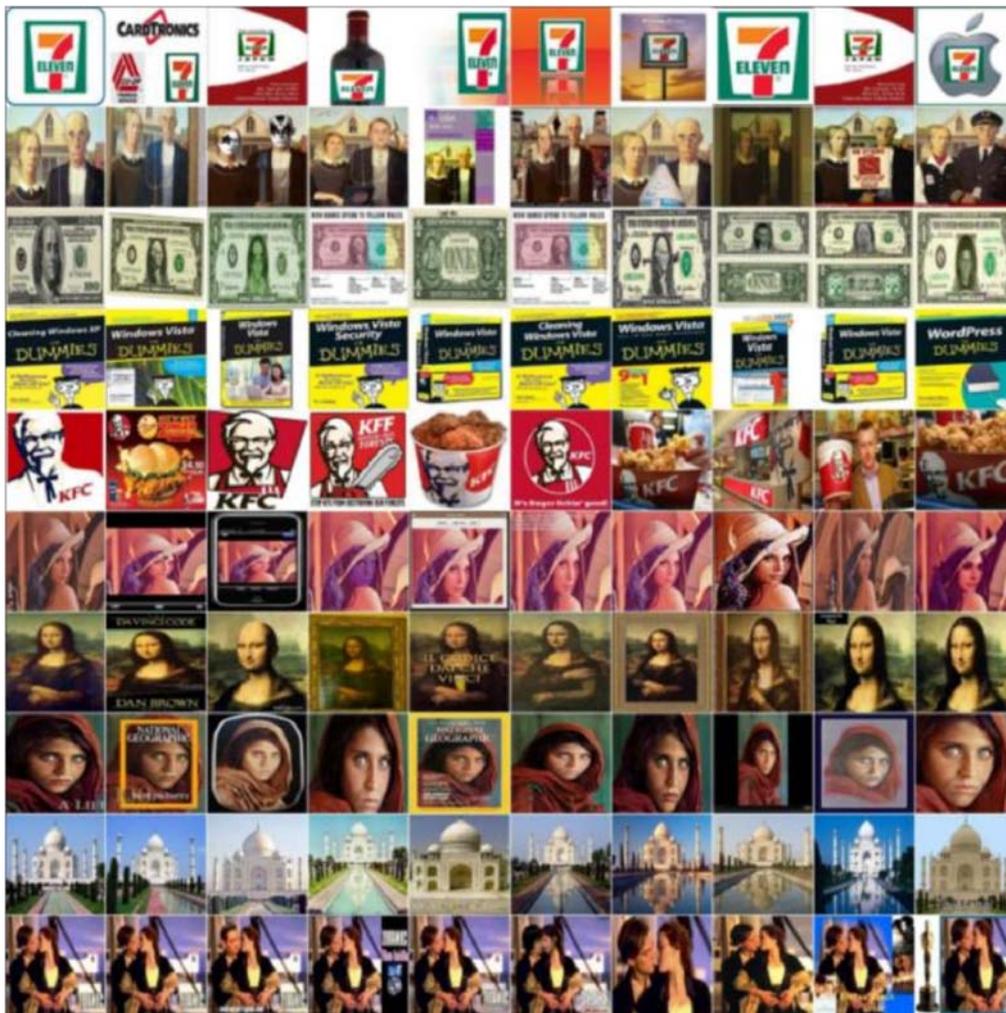


FIG. 3. DATASET USED FOR FEATURES MATCHING. FIRST COLUMN FROM THE LEFT CONTAINS THE QUERY IMAGES, AND REST ALL ARE GRADUAL DEFORMED TRANSFORMATIONS

IR-II Dataset



(a) IMAGE SELECTED RANDOMLY FROM EVERY SCENE



(b) 10 DISTORTED COPIES OF IMAGES ARE SELECTED RANDOMLY FROM DIFFERENT SCENES

FIG. 4. IR DATASET EMPLOYED FOR EXPERIMENT OF IMAGE RETRIEVAL. THE DATASET HAS 1K IMAGES OF 36 DIFFERENT SCENES. THE TRANSFORMATION INCLUDE THE TRANSFORMATIONS MENTIONED IN TABLE 1

## 4.1 Evaluation Metrics

Recall against 1-precision are used as evaluation metrics, as quantity of false point correspondences relative to total quantity of correspondences are efficiently expressed by 1-precision [10], similar matrices are widely used for descriptors evaluation [4,9,10], recall and 1-precision are determined as below:

$$\text{recall} = \frac{\text{Number of True Positives}}{\text{Total Positives}}$$

$$1 - \text{Precision} = \frac{\text{Number of False Negatives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

False negatives and true positives are determined with the overlap error [4,45], overlap error determines the efficiency of affine regions correspondences under homography (transformation). Details can be found in [4,45].

## 4.2 Descriptors Matching

We used nearest neighbor matching strategy between the descriptors. Let  $D_Q$  and  $D_R$  be the set of descriptors from image Q and R, where  $R = \{L_1, L_2, \dots, L_5\}$ , the point pair  $(d_i^Q, d_j^R)$  is considered a match if subsequent two conditions hold. The Euclidean distance E, where  $E(d_i^Q, d_j^R) = \min_{d_k^R \in D_R} E(d_i^Q, d_k^R)$  and following inequality holds:  $E(d_i^Q, d_j^R) \times T_m \leq \min_{d_k^R \in D_R, k \neq j} E(d_i^Q, d_k^R)$  where  $T_m$  is threshold for consistent matching. The values of precision and recall are obtained by the changing the  $T_m$ .

## 5. RESULTS

In this section, we provide the evaluation results of C-CSLBP. For different transformations, CSLBP gives better performance compared to SIFT, the main limitation of CSLBP is its dimensions which is quite often the double of SIFT. All values in CSLBP descriptor are floating points that causes CSLBP to consume more memory than SIFT, and SIFT is also faster for features matching. The C-CSLBP is a complimentary approach to CSLBP that has equal dimensions compared to SIFT, and the robustness is approximately similar to CSLBP.

When we perform C-CSLBP compression in the CSLBP descriptor, we assume that when CSLBP and C-CSLBP descriptors are extracted from the similar points, the distances among resultant C-CSLBP descriptors are correlated with the distances among CSLBP descriptors. If C-CSLBP preserves the relative ordering of distances among CSLBP descriptors pairs, considering similar circumstances we expect C-CSLBP to provide nearest neighbor matches similar to CSLBP. We validate this hypothesis through experiment. In our experiment, we take two images of a scene from the Mikolajczyk dataset and extract DoG keypoints from the images, then according to Section 4.2 algorithm, we acquire both CSLBP and C-CSLBP matches for these two images, and after that, according to the two algorithms, comparison between correspondences is analyzed. In an image pair, considering every keypoint of the first image, if we find a keypoint match in the second image according to CSLBP as well as C-CSLBP, we employ a scatter plot to plot the point  $(x,y)$  where  $x$  and  $y$  are the indexes of the point matching according to CSLBP and C-CSLBP, respectively. For a consistent match, we would have  $x = y$ . As a result, the diagonal scatter plot is obtained for a completely consistent descriptor.

The results of this method for an image Q matched with  $L_1$  (least distorted) above all six transformations are illustrated in Fig. 5. This can be noticed that, for all sorts of transformations, C-CSLBP is rather consistent with CSLBP.

### 5.1 Features Matching Performance

The proposed descriptor performs equivalently better on many challenging transformations. For each transformation, the query image Q is compared with remaining five degraded images  $\{L_1, L_2, \dots, L_5\}$ . For various transformations, CSLBP performance is better than SIFT

such as Illumination change, image blur, and JPEG compression. Whereas, for viewpoint variation and rotation types of transformation, the CSLBP has approximately the same performance compared to SIFT as illustrated in Fig. 6. The performance achieved by C-CSLBP is similar to CSLBP. The MSE (Mean Square Error) after compression of CSLBP is very less as shown in Fig. 7(b) (Left), precision on different values of recall are calculated for both CSLBP and C-CSLBP, and the average of their squared precision differences on different transformations are shown. It can be seen that C-CSLBP have minimum error after compression. Fig. 7 (Right) show the average MSE of CSLBP and C-CSLBP when compared

to perfect descriptor, perfect descriptor can be interpreted as having F-score equals to one on every type of transformations.

For each transformation, matching performance of query image Q with  $L_1$  and  $L_3$  are reported, it can be seen that C-CSLBP has approximately the same performance compared to CSLBP, SIFT and PCA-SIFT. In case of JPEG transformations, the performance is same on initial levels but get lower gradually. The performance of C-CSLBP is better than PCA-SIFT for blur, illumination, viewpoint change, and rotation types of transformations.

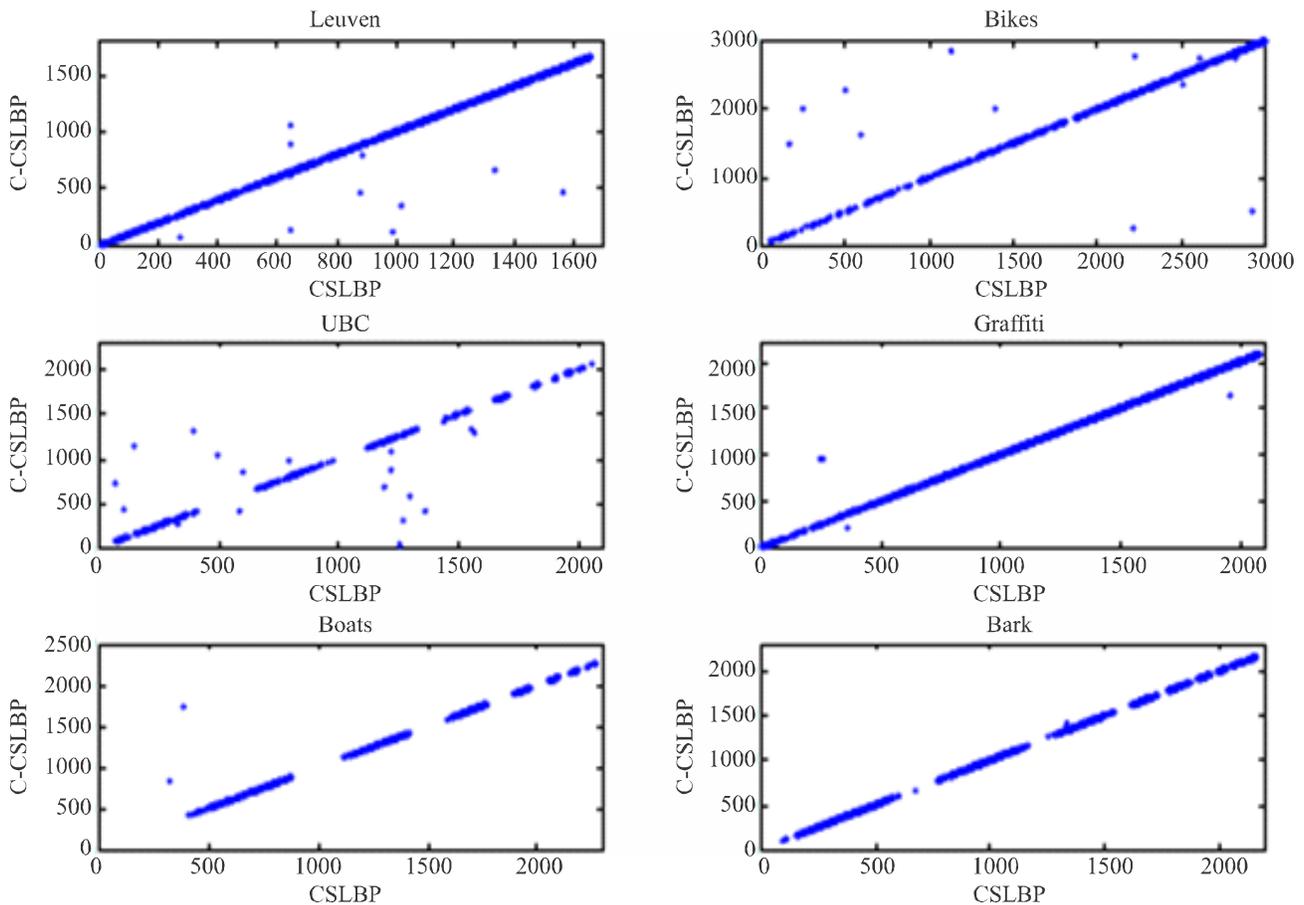


FIG. 5. SCATTER PLOTS OF MATCH INDICES BY C-CSLBP AND CSLBP UNDER VARIOUS TRANSFORMATIONS. EVERY POINT REPRESENTS A KEYPOINT IN IMAGE-1. THE X AND Y COORDINATES OF THE POINT ARE THE INDICES OF ITS MATCH IN IMAGE-2 ACCORDING TO CSLBP AND C-CSLBP, RESPECTIVELY. AN ACCURATE REPLICAS OF CSLBP CAN OBTAIN A DIAGONAL PLOT. THE POINTS THAT WERE NOT MATCHED BY BOTH CSLBP AND C-CSLBP ARE INDICATED AS GAPS

### 5.2 Utility of C-CSLBP for Image Retrieval

For the evaluation of C-CSLBPs retrieval performance aimed at image retrieval, we employ mAP (mean average precision). To acquire mAP, match score for each gallery image is obtained for every query image. Then match score is ranked and match score cutoff is found. The match score cutoff is employed to retrieve all the replicas of the query image. Then, the retrieval precision is computed at that cutoff (dividing the total number of copies by the total number of images in gallery that achieve match score above the threshold). At the end, the precision value over each query image is averaged.

For a pair of images  $I_p$  and  $I_q$ , the match score is, provided a value to threshold of the stable matching ratio  $T_m$ ,

$$M(I_p, I_q; T_m) = \frac{\|D(I_p) \cap^{T_m} D(I_q)\|}{\|D(I_p)\|}$$

where  $D(I_p)$  and  $D(I_q)$  represent the descriptors sets in corresponding images  $I_p$  and  $I_q$ , respectively.  $D(I_p) \cap^{T_m} D(I_q)$  represent the paired descriptors set in  $D_p$  and  $D_q$  that are found to be in correspondence by employing stable matching ratio threshold  $T_m$ .  $\|D(I_p)\|$  represent the quantity of keypoints in image  $I_p$ . Ideally, the earliest entries of the ranked list should be the image itself followed by distorted versions of  $I_p$  in  $M$ .

We provide our results under two different circumstances: In one circumstance, when the value of  $T_m$  is presented at the standard value of 1.5 (default value employed by several implementations of SIFT like VLFEAT), and in other circumstance when the value of  $T_m$  is tuned on the dataset to achieve the finest probable outcome. We altered the value of  $T_m$  between 1.0 and 3.5 for both every keypoint and descriptor. We acquired the pair wise match scores, and after that, based on match score, we sorted the results for every query image.

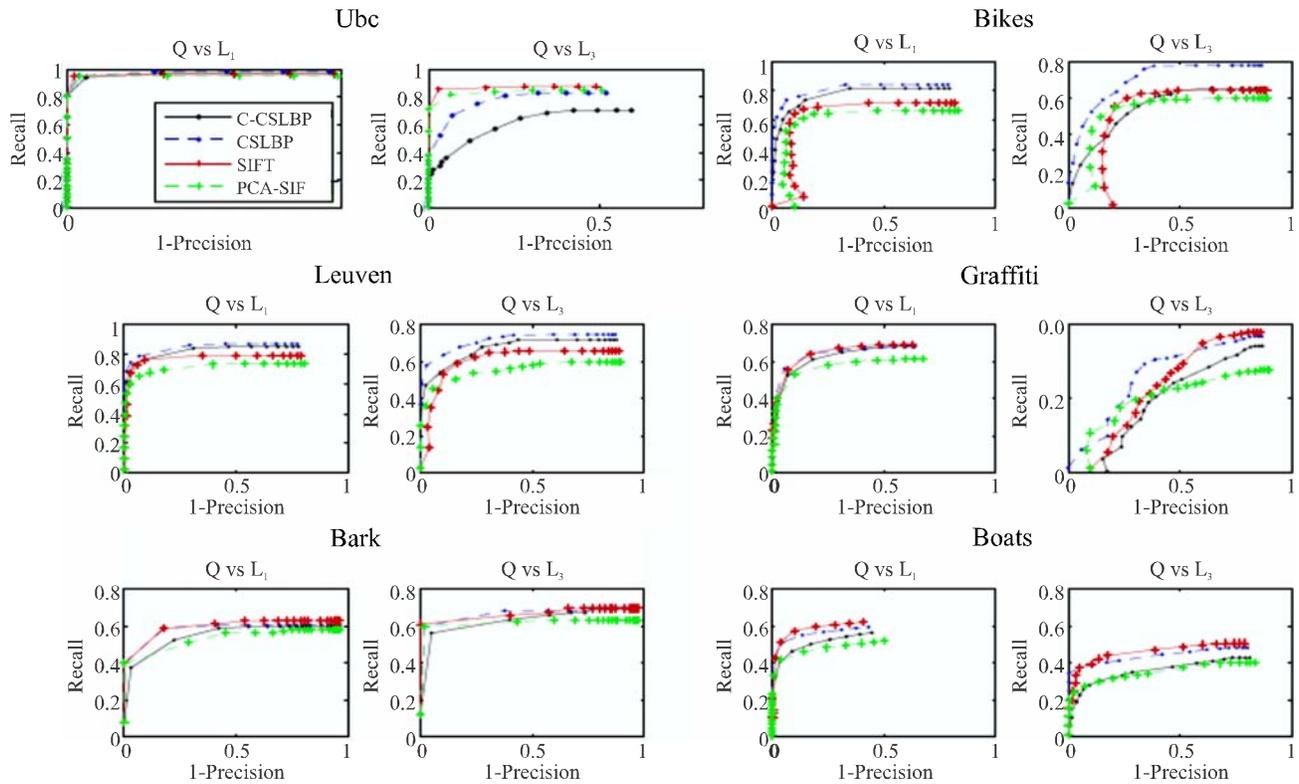


FIG. 6. MATCHING PERFORMANCE OF C-CSLBP COMPARED TO CSLBP, SIFT AND PCA-SIFT

Table 2 shows the mAP of proposed compression on IR dataset. C-CSLBP has approximately same performance compared to CSLBP (before compression). C-CSLBP retains not only robustness (Fig. 6) but also distinctiveness (Table 4).

### 5.3 Memory Consumption

C-CSLBP is 128-length descriptor; it achieves significant gain in memory compared to CSLBP. The CSLBP descriptor requires 256x8 bytes to store one descriptor, whereas C-CSLBP descriptor requires only 128 bytes of memory to store one descriptor. In case of Object Classes Challenge 2010 dataset that have 11321 images and 6.5 million keypoints using Difference of Gaussian detectors. To store 6.5 million descriptors statistically, 12.4 GB memory is required by CSLBP descriptors, whereas only 0.8 GB memory required by C-CSLBP descriptors as shown in Fig. 8.

TABLE 2. MEAN AVERAGE PRECISION ON IR DATASET

Detector	Tm = 1.5		Tuned Tm	
	CSLBP	C-CSLBP	CSLBP/Tm	C-CSLBP/Tm
Harris Affine	0.65	0.62	0.68/1.7	0.66/1.45

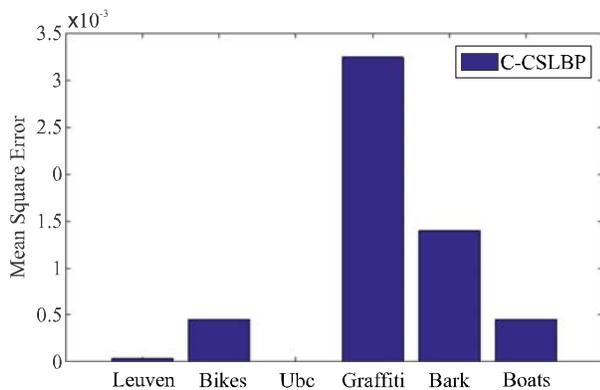


FIG. 7. MSE OF C-CSLBP AND CSLBP. LEFT SHOWS THE MSE OF C-CSLBP W.R.T THE PERFORMANCE OF CSLBP. RIGHT SHOWS THE MSE OF C-CSLBP AND CSLBP W.R.T THE IDEAL PERFORMANCE BY CONSIDERING AVERAGE F-SCORE TO ONE

## 6. CONCLUSION

We have proposed an efficient and straightforward compression of CSLBP without applying any dimensionality reduction techniques. We achieve similar performance with reduced dimensions. The C-CSLBP outperforms SIFT as well as PCA-SIFT under illumination and blur transformations. As a result of the compact length of proposed descriptor, it is efficient for features matching and memory storage compared to CSLBP and SIFT. Since, C-CSLBP is textured based descriptor; it can efficiently be used for smart phones and tablets.

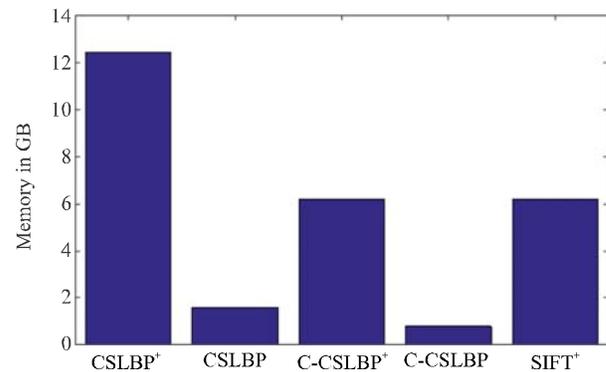
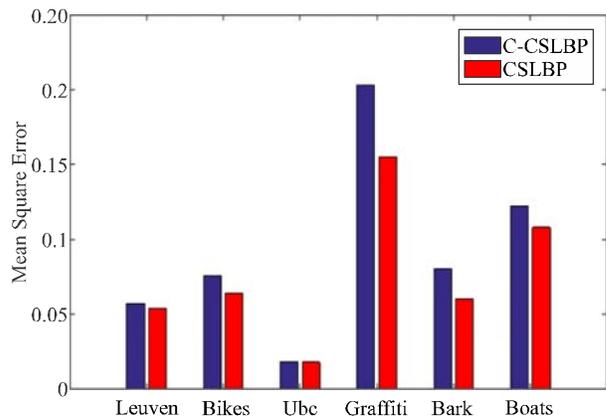


FIG. 8. MEMORY REQUIRED TO STORE 6.5X10<sup>6</sup> DESCRIPTORS (IN GIGABYTES). THE + INDICATES THAT THE DESCRIPTOR LENGTH IS NORMALIZED TO UNIT LENGTH



## ACKNOWLEDGMENTS

Authors are thankful to National Institute of Informatics, Tokyo, for supporting the research. Authors are also thankful to Dr. Akhtar Hussain, for his technical support in paper writing.

## REFERENCES

- [1] Lowe, D.G., "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, Volume 60, pp. 91-110, 2004.
- [2] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L., "Speeded-Up Robust Features (SURF)", *Computer Vision Image Understanding*, Volume 110, No. 3, pp. 346-359, 2008.
- [3] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., "ORB: An Efficient Alternative to SIFT or SURF", *International Conference on Computer Vision*, November, 2011.
- [4] Mikolajczyk, K., and Schmid, C., "A Performance Evaluation of Local Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 27, No. 10, pp. 1615-1630, October, 2005.
- [5] Heinly, J., Dunn, E., and Frahm, J.M., "Comparative Evaluation of Binary Features", *European Conference on Computer Vision*, *Lecture Notes in Computer Science*, pp. 759-773, 2012.
- [6] Boiman, O., Shechtman, E., and Irani, M., "In Defense of Nearest-Neighbor Based Image Classification", *Computer Vision and Pattern Recognition*, 2008.
- [7] Snavely, N., Seitz, S.M., and Szeliski, R., "Skeletal Sets for Efficient Structure from Motion", *Computer Vision and Pattern Recognition*, 2008.
- [8] Douze, M., Jegou, H., and Schmid, C., "An Image-Based Approach to Video Copy Detection with Spatio-Temporal Post-Filtering", *IEEE Transactions on Multimedia*, Volume 12, No. 4, pp. 257-266, June, 2010.
- [9] Heikkila, M., Pietikainen, M., and Schmid, C., "Description of Interest Regions with Local Binary Patterns", *Pattern Recognition*, Volume 42, pp. 425-436, 2009.
- [10] Ke, Y., and Sukthankar, R., "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors", *Computer Vision and Pattern Recognition*, pp. 511-517, 2004.
- [11] Yamazaki, M., and Fels, S., "Local Image Descriptors Using Supervised Kernel ICA", *IEICE Transactions on Information and Systems*, pp. 1745-1751, 2009.
- [12] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., "BRIEF: Binary Robust Independent Elementary Features", *European Conference on Computer Vision*, September, 2010.
- [13] Leutenegger, S., Chli, M., and Siegwart, R., "BRISK: Binary Robust Invariant Scalable Keypoints", *International Conference on Computer Vision*, 2011.
- [14] Ambai, M., and Yoshida, Y., "CARD: Compact and Real-Time Descriptors", *International Conference on Computer Vision*, 2011.
- [15] Indyk, P., and Motwani, R., "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality", pp. 604-613, 1998.
- [16] Liu, X., He, J., and Lang, B., "Multiple Feature Kernel Hashing for Large-Scale Visual Search", *Pattern Recognition*, Volume 47, No. 2, pp. 748-757, 2014.
- [17] Wang, J., Kumar, S., and Chang, S.F., "Semi-Supervised Hashing for Large-Scale Search", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 34, No. 12, pp. 2393-2406, 2012.
- [18] Strecha, C., Bronstein, A., Bronstein, M., and Fua, P., "LDA Hash: Improved Matching with Smaller Descriptors", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 34, No. 1, pp. 66-78, 2012.
- [19] Philbin, J., Chum, O., Isard, M., Sivic, J., and Zisserman, A., "Object Retrieval with Large Vocabularies and Fast Spatial Matching", *Computer Vision and Pattern Recognition*, 2007.
- [20] Ling, H., Cheng, H., Ma, Q., Zou, F., and Yan, W., "Efficient Image Copy Detection using Multiscale Fingerprints", *IEEE Transactions on Multimedia*, 2012.
- [21] Boiman, O., Shechtman, E., and Irani, M., "In Defense of Nearest-Neighbor Based Image Classification", *Computer Vision and Pattern Recognition*, 2008.
- [22] Zhou, W., Lu, Y., Li, H., and Tian, Q., "Scalar Quantization for Large Scale Image Search", *ACM International Conference on Multimedia*, pp. 169-178, 2012.

- [23] Baber, J., Dailey, M., Satoh, S., and Bakhtyar, M., "BIG-OH: Binarization of Gradient Orientation Histograms", *Image and Vision Computing*, 2014.
- [24] Baber, J., Satoh, S., Afzulpurkar, N., and Bakhtyar, M., "Q-CSLBP: Compression of CSLBP Descriptor", *LNCS Processing PCM*, 2012.
- [25] Heikkilä, M., and Pietikainen, M., "A Texture-Based Method for Modeling the Background and Detecting Moving Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 28, No. 4, pp. 657-662, 2006.
- [26] Ojala, T., Pietikainen, M., and Maenpää, T., "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 24, No. 7, pp. 971-987 2002.
- [27] Xiaoyu, W., X., H.T., and Shuicheng, Y., "An HOG-LBP Human Detector with Partial Occlusion Handling", *International Conference on Computer Vision*, pp. 32-39, 2009.
- [28] Ahonen, T., Hadid, A., and Pietikainen, M., "Face Description with Local Binary Patterns: Application to Face Recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 28, No. 12, pp. 2037-2041, 2006.
- [29] Pietikainen, M., Nurmela, T., Maenpää, T., and Turtinen, M., "View-Based Recognition of Real-World Textures", *Pattern Recognition*, Volume 37, No. 2, pp. 313-323, 2004.
- [30] Rosten, E., and Drummond, T., "Machine Learning for High-Speed Corner Detection", *European Conference on Computer Vision, Lecture Notes in Computer Science*, pp. 430-443, 2006.
- [31] Tola, E., Lepetit, V., and Fua, P., "DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 32, No. 5, pp. 815-830, 2005.
- [32] Ta, D.N., Chen, W.C., Gelfand, N., and Pulli, K., "SURFTrac: Efficient Tracking and Continuous Object Recognition Using Local Feature Descriptors", *Computer Vision and Pattern Recognition*, 2009.
- [33] Baber, J., Afzulpurkar, N., Dailey, M., and Bakhtyar, M., "Shot Boundary Detection from Videos Using Entropy and Local Descriptor", *International Conference on Digital Signal Processing*, 2011.
- [34] Baber, J., Afzulpurkar, N., and Satoh, S., "A Framework for Video Segmentation Using Global and Local Features", *International Journal of Pattern Recognition and Artificial Intelligence*, Volume 27, No. 5, 2013.
- [35] Baber, J., Afzulpurkar, N., and Bakhtyar, M., "Video Segmentation into Scenes Using Entropy and SURF", *International Conference on Emerging Technologies*, pp. 1-6, 2011.
- [36] Liu, W., Wang, J., Ji, R., Jiang, Y.G., and Chang, S.F., "Supervised Hashing with Kernels", *Computer Vision and Pattern Recognition*, pp. 2074-2081 37. Weiss, Y., Torralba, A., and Fergus, R., *Spectral hashing*, NIPS, 2008
- [37] Trzcinski, T., and Lepetit, V., "Efficient Discriminative Projections for Compact Binary Descriptors", *European Conference on Computer Vision*, 2012.
- [38] Sivic, J., and Zisserman, A., "Video Google: A Text Retrieval Approach to Object Matching in Videos", *International Conference on Computer Vision*, 2003.
- [39] Nister, D., and Stewenius, H., "Scalable Recognition with a Vocabulary Tree", *Computer Vision and Pattern Recognition*, Volume 2, pp. 2161-2168, 2006.
- [40] Jegou, H., Douze, M., and Schmid, C., "Product Quantization for Nearest Neighbor Search", *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Volume 33, No. 1, pp. 117-128, 2011.
- [41] Zhou, W., Lu, Y., Li, H., Song, Y., and Tian, Q., "Spatial Coding for Large Scale Partial Duplicate Web Image Search", *International Conference on Multimedia*, 2010.
- [42] Haralick, R.M., Shanmugam, K., and Dinstein, I., "Textural Features for Image Classification", *IEEE Transactions on Systems, Man and Cybernetics*, 1973.
- [43] Calonder, M., Lepetit, V., Strecha, C., and Fua, P., "BRIEF: Binary Robust Independent Elementary Features", *European Conference on Computer Vision*, pp. 778-792, 2010.
- [44] Mikolajczyk, K., and Schmid, C., "Scale and Affine Invariant Interest Point Detectors", *International Journal of Computer Vision*, Volume 60, pp. 63-86, 2004.