Learning Local Image Descriptors Using Binary Decision Trees

Juha Ylioinas, Juho Kannala, Abdenour Hadid, and Matti Pietikäinen
Center for Machine Vision Research
University of Oulu
firstname.lastname@ee.oulu.fi

Abstract

In this paper we propose a unified framework for learning such local image descriptors that describe pixel neighborhoods using binary codes. The descriptors are constructed using binary decision trees which are learnt from a set of training image patches. Our framework generalizes several previously proposed binary descriptors, such as BRIEF, LBP and their variants, and provides a principled way to learn new constructions which have not been previously studied. Further, the proposed framework can utilize both labeled or unlabeled training data, and hence fits to both supervised and unsupervised learning scenarios. We evaluate our framework using varying levels of supervision in the learning phase. The experiments show that our descriptor constructions perform comparably to benchmark descriptors in two different applications, namely texture categorization and age group classification from facial images.

1. Introduction

Image description is a key component in almost all computer vision applications concerning detection, recognition and classification. Among the main requirements for a descriptor are discriminative efficiency and robustness against degradations due to many factors such as noise and varying imaging conditions. Classical methods such as Local Binary Pattern (LBP) [15] and Scale-Invariant Feature Transform (SIFT) [11] have been widely examined and have gained credits due to their robustness against challenges such as pose and illumination variation.

The latest wave of development has seen the birth of local descriptors that are quick to compute and compact in their representation. A representative of this style of descriptors is so called Binary Robust Independent Elementary Features (BRIEF). The motivation for developing fast and compact description methods has risen partly due to the ever increasing data and the proliferation of hand-held devices for which the computational lightness is always vital. For SIFT, for example, frequently mentioned drawbacks are the computational cost of the feature calculation and slow nearest-neighbor matching. Furthermore, although LBP is a very fast in terms of feature computation it often leads to very long representations. Dimension reduction methods such as Principal Component Analysis (PCA) can help to speed up matching while using SIFT or LBP but at the cost of added computational load and sometimes even with a dropped matching accuracy.

Besides speed and compactness another essential requirement for the construction of image descriptors is that they are discriminative. A question then arises about how to simultaneously achieve these all appealing properties. By operating on simple pixel difference tests one can easily be convinced about the speed. Compactness, in turn, can be achieved by a descriptor operator that divides the resulting descriptor or feature space as evenly as possible with respect to all possible sample occurrences. Finally, to have a discriminative description the feature space should be such that all the regions collect as much as possible sample occurrences from a particular class and are only slightly mixed of possible classes occurring. Clearly, to achieve all of these properties powerful learning methods are needed.

In this paper, we propose a unified framework for learning local image descriptors using binary decision trees. In fact, it is clear and well known that many previously proposed descriptors, which represent image patches using binary codes, such as LBP [15], LPQ [16] and BRIEF [3], can be computed by evaluating a sequence of binary-valued functions. However, we suggest to view such descriptor constructions as a special case of a more general decision tree model where the binary code of an image patch is obtained by assigning the patch to one of the leaf nodes. That is, a binary-valued function of pixel intensities is computed at each node of the tree so that its value determines the node evaluated at the next level, thereby defining a unique route from the root node to one of the leaf nodes encoded as a bit string. In this kind of a setting, the aforementioned traditional image descriptors (e.g. LBP, LPQ, BRIEF) correspond to a tree in which each node at a particular depth evaluates the same function. This constraint makes the function evaluations independent of each other and therefore the tree structure can be reduced to a so-called *fern* [17]. Yet, considering these previous image descriptors as special cases of more general decision trees opens up possibilities for potentially useful generalizations and for utilizing the rich theory and practice of decision trees [5] for learning new descriptors.

There are a few previous works which have used decision trees in the context of local image description. For example, [10] used randomized trees for keypoint classification in wide baseline image matching, [20] used randomized trees for image categorization and segmentation, and [13] learned local binary pattern features for description of face regions using a decision tree. However, almost without an exception, all of these previous works, as well as most other related works, use decision trees in the supervised setting where labeled training images are available. In contrast, we take the unsupervised setting into deeper investigation by additionally considering the possibility of using decision trees to learn generic image descriptors using unlabeled non-application-specific image data. In fact, inspired by [9], one of our goals is to construct such descriptors that do not need to be trained specifically for each application and which could be used as off-the-shelf alternatives for conventional hand-crafted descriptors, like BRIEF or LBP. To the best of our knowledge, decision trees have not been previously used for learning general-purpose binary descriptors directly from raw pixel values.

Finally, it should be noted that besides pointing out connections between different previous local descriptors and providing ways for learning new constructions, our framework responds well to practical requirements since binary decision trees provide compact and discriminative features which are fast to compute. Moreover, as our approach can utilize both labeled and unlabeled data it fits well to different learning scenarios and also has potential for learning the characteristics of different application domains if additional training data is available.

2. Related work

Statistical histogram over a discrete vocabulary of local texture features is an effective way of image description. In this paradigm, the responses to a descriptor of local image patches are assigned to predefined bins according to some partition of the feature space.

One of the pioneering works under the visual vocabulary paradigm is Local Binary Patterns (LBP) [18]. LBP uses a dataset-independent dictionary of local features which are based on simple pixel intensity comparisons on a local neighborhood. The output of the LBP operation is a binary code string which characterizes texture properties within this region. The binary strings are then mapped to their decimal counterparts and collected into a histogram which is used as image description. The simplest form of LBP operates on image pixels by thresholding their 3×3 neighborhood with the center value, but later it was extended to use neighborhoods of different sizes with different sampling geometries such as circular and elliptical (bilinearly interpolating values at non-integer pixel coordinates) which have both proven to work reasonably well in a vast amount of applications. The most important characteristic of LBP is its robustness to monotonic gray-scale changes caused, for example, by illumination variations. This is achieved by the fact that LBP operates on the signs of pixel value differences throwing away the information about absolute pixel magnitudes.

Binary Robust Independent Elementary Features (BRIEF) [3] shares the idea of LBP to form a descriptor which is based on pixel comparisons on a local neighborhood. The BRIEF descriptor operates on image patches by a predefined set of n (x, y)-location pairs that define the set of binary tests. The output of the BRIEF operation is a binary string.

Besides being both robust against monotonic intensity changes and shown to be highly discriminative, LBP and BRIEF are computationally simple and very fast to extract. However, both of them share the same question about how to select the binary tests used in feature calculation. Compared with BRIEF, there are fewer possibilities to select the binary tests in LBP as the center is always fixed. Further, for LBP, the problem is even more relaxed, as developers usually fix the sampling geometry, such as a ring or an ellipse, and then pick equally sampled points around that. For BRIEF, authors in [3] tested different kinds spatial arrangements by selecting the test locations according to uniform and Gaussian distributions. Although both of these methods have shown to work well, one might ask whether there are more sophisticated methods available than handcrafting the sampling points or randomly deriving them according to some distribution.

After fixing the sampling strategy, a popular way to proceed is then to select the most significant patterns or pools of patterns in this topology-defined feature space. For LBP, these methods cover different feature selection techniques such as the widely used *uniform* and *rotation invariant* patterns. Different kinds of heuristical search strategies, such as beam search [12] or Sequential Floating Forward Selection (SFFS) [1], have been also tried but without any significant improvements. One must remember that feature selection methods are always sub-optimal, due to the number of possible subspaces which is $\binom{D}{d}$ where d is the dimension of the subspace. Also, the options for BRIEF have not yet been studied in-depth. Instead of randomly deriving binary tests according to some distribution, a greedy learning algorithm was proposed to find a subset of uncorrelated binary tests that provide high variance [19].

To tackle the raised sampling and feature selection issues, a method called Local Quantized Patterns (LQP) [7] was proposed. LQP was introduced as a generalization of local pattern features making use of vector quantization and letting to have many more pixels and quantization levels without the loss of simplicity and computational efficiency. One of the main objectives of LQP is to get rid of handcrafted sampling topologies by densely sampling larger local pattern neighbourhoods and then using K-means to cluster the resulting set of binary patterns off-line to build a very large look-up table which later allows run-time coding of local patterns. LQP is partly inspired by the famous visual words approach where a set of training images are convolved with a filter bank to generate filter responses. Exemplar filter responses, found by K-means clustering, are then selected as *textons* for labeling each filter response and finally every pixel.

A kind of an opposite method to LQP was proposed in [13] where decision trees were introduced to learn the most discriminative set of neighborhood pixels for intensity value comparisons. The observation behind this method, called Decision Tree LBP (DT-LBP), was the operation of an LBP over a given neighborhood which was stated equivalent to the operation of a fixed binary decision tree. Thus, instead of using a fixed binary decision tree, the paper proposes to use decision tree learning for finding out the most discriminative pixel value comparisons. The immediate aim of the DT-LBP method is to produce compact and discriminative descriptors without such a deep pattern mining and later clustering present in LQP.

In [17], a concept called decision fern was introduced as a specific case of a tree structure and with an application to image patch processing for keypoint recognition. A fern is formed from a tree by first constraining it to systematically perform the same test across each hierarchy level, which then results in the same test of the path taken to get to a particular node. As a result, ferns do not contain hierarchical structure but apply a linear sequence of tests. In [17], ferns were used as replacements for trees based on the argument that the tree structure itself was not the key factor in successful patch recognition but rather combining the groups of binary tests. It is evident, by definition, that the LBP descriptor can be seen as a kind of a decision fern.

In their most original form, LBP and BRIEF descriptors rely on simple pixel value comparisons describing the result of a set of these operations by zeros and ones. For LBP, many studies propose various kinds of sampling geometries for characterizing the local neighborhood. Some was shown to work better than the basic rectangular or circular LBP but only few put a serious attention on the question why did the topology at hand perform better than the basic one. LQP was introduced to tackle the problem of choosing the sampling geometry by proposing a descriptor operating on a densely sampled patch of a size 5×5. However, as also noted by the authors [7], using larger neighborhoods such as 7×7 or even bigger ones, the developer is again forced to handcraft sampling geometries as the dense sampling in this case yields a lot more than 24 binary tests which was stated to be an upper limit for implementation in hardware.

For BRIEF, there are only few methods for selecting the best set of pixel value comparisons. Moreover, to the best of our knowledge, BRIEF has never been experimented as a dense descriptor but rather as a method for sparse keypoint point description. Partly towards these ends, we next introduce a framework for learning disriminative and compact descriptors that are based on binary decision trees. Although sharing many of the ideas already presented in [13], [14], and [17] our method can be seen as a generalization of all these works. We evaluate our framework using decision trees under the dense image description mode meaning the final description is formed from the whole image or at least from major parts.

3. Learning local binary descriptors

When using binary descriptors like LBP or BRIEF a good starting point is to clarify the underlying mechanism and the efficiency they are based on. At first, thresholding pixel value differences is an efficient coding per se, as it makes the descriptor invariant to monotonic gray-scale changes. What follows is then the selection of the best set of pixels for the comparisons. For LBP, while using circular or elliptical sampling geometries this set of pixels is always fixed. This is motivated by the patterns that occur in textures and that the most interesting ones are more or less extractable by using the given geometries.

Because each pixel value thresholding operation included in a binary descriptor is responsible for such a critical task as feature space partition, it is desirable to find the most optimal set of those. In LBP calculation, however, this does not apply as the pixels are chosen based on the fixed geometry. To tackle this, Maturana *et al.* proposed to use decision tree learning algorithms to find the most discriminative set of pixel comparisons [13]. Although they showed the method improves standard LBP calculation in face recognition, their framework was still highly tuned for operating on faces. First, they trained many different tree descriptors based on different face areas and did not try to explain in depth what really happens behind the curtains.

The main goal of our framework is to gain ability to learn discriminative and compact descriptors by first sampling a representative yet reasonable number of training image patches to represent the feature space and then learning the optimal parition of this feature space using decision trees. It is of further interest to evaluate whether it is possible to learn general purpose descriptors using natural images as it was done in [9]. Further, the aim is not to put any restrictions on finding the best set of pixel comparisons, but to generalize the descriptor to operate like the BRIEF descriptor. Following the steps presented in [13] and [14], the algorithm for learning tree-based local binary descriptors is given in Alg.1.

Algorithm 1 $Tree(L)$

```
**Input:** A set of training image patches $L$
  **if** $max\_depth == true$ **then**
    **return**
  **else**
    $(f_t, \theta_t) \leftarrow chooseTest(L)$
    $L_l \leftarrow \{(\mathbf{x}_i, y_i) \in L_t | f_t(\mathbf{x}_i) \geq \theta_t\}$
    $L_r \leftarrow \{(\mathbf{x}_i, y_i) \in L_t | f_t(\mathbf{x}_i) < \theta_t\}$
    **return** $f_t$, $\theta_t$, $Tree(L_l)$, $Tree(L_r)$
  **end if**
```

Algorithm 2 $chooseTest(L)$

```
**Input:** A set of training image patches $L$
**Output:** A binary test $f$ and a threshold $\theta$
  evaluate n possible binary tests
  **for** $j = 1$ to $n$ **do**
    **scores** $\leftarrow evaluate(L, f_j, \theta_j)$
  **end for**
  find the index of the best test
  $j^* = \arg\max_j$ **scores**
  **return** $f_{j^*}, \theta_{j^*}$
```

The binary descriptor is learned recursively top-down. The training phase starts by defining a complete descriptor space corresponding to a training set $L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ in which $\mathbf{x}_i \in R^{w \times h}$, w and h defining the size of the patches. The recursive algorithm evaluates each node t by dividing L into two distinct subsets L_r and L_l so that $L_r \cup L_l = L$ and $L_r \cap L_l = \emptyset$. This is achieved by the $chooseTest$ function which ranks all possible binary tests with respect to L given the spatial support of the descriptor defined by $w \times h$. The $chooseTest$ function can be based on for example pixel pairs, trios, or on any kind of function of pixels. If the test is fixed as a sign of the difference between a pixel pair (like in our experiments) the function is defined as $f(\mathbf{x}_i) = sgn(x_{i_a} - x_{i_b})$, where a and b denote the pixel positions in the patch. In this kind of setting, the number of possible tests is given by the size of the patch $\mathbf{x}_i$ so that there are altogether $\binom{w \times h}{2}$ tests that could be evaluated. At some point, especially while using larger image patches, the number of possible tests may grow too high so that the evaluation must be based on random subsets of all possible tests. Also, several different threshold values θ can be evaluated, but one must remember that if the threshold is not fixed the number of possible binary tests grows further.

The selection between unsupervised and supervised learning then defines what is maximized. In unsupervised scenario where class information is not utilized one can use the Shannon's split entropy which is defined as

$$H(L) = -\sum_{p=1}^{2} \frac{|L_p|}{|L|} \log_2 \frac{|L_p|}{|L|}, \qquad (1)$$

where $|\cdot|$ returns the size of the set and p stands for the partition. The maximum is reached when both of the partitions have equal number of training patches.

In supervised scenario, where the class information is utilized, pixel value comparisons are ranked based on the information gain of the split, defined as

$$I(L) = H_C(L) - \sum_{p=1}^{2} \frac{|L_p|}{|L|} H_C(L_p), \qquad (2)$$

where H_C is defined as

$$H_C(L) = -\sum_{c \in \boldsymbol{C}} \frac{|L_c|}{|L|} \log_2 \frac{|L_c|}{|L|}, \qquad (3)$$

where $|L_c|$ is the number of patches belonging to class c.

The $Tree$ function is recursively called until it reaches the maximum depth which is experimentally set. As it can be seen, the main parameters of the overall process are the desired level of supervision, the size of the descriptor's spatial support and the maximum depth. In our experiments, the spatial support is fixed as square $S \times S$, varying S between 3, 5, and 7. In general, however, there are no limitations for using any rectangular or any other geometrical shape for the neighborhood. In our experiments, the depth is varied from 5 to 10.

For fern-like structures, the root node test is selected as previously, but after that we must modify the given split entropy and information gain so that both are functions of all nodes at a depth l. Thus, if the root note is indexed as 0 and the rest in ascending order from left to right, the split entropy, starting from the depth $l = 1$, is defined as follows

$$H_l(L) = -\sum_{j=2^l-1}^{2^{l+1}-2} \frac{|L_j|}{|L|} \log_2 \frac{|L_j|}{|L|}, \qquad (4)$$

where j stands for the node index at a depth of l. Further, the information gain is given as

$$I_l(L) = H_C(L) - \sum_{j=2^l-1}^{2^{l+1}-2} \frac{|L_j|}{|L|} H_C(L_j), \qquad (5)$$

where j and l are as in (4), and $H_C(\cdot)$ as in (3). The test that gives the highest split entropy or information gain is then selected as the $(l+1)$th component for the linear sequence of the fern structure.

4. Experimental analysis

We evaluate the performance of the proposed descriptor learning framework by conducting experiments on two recognition problems: texture categorization and biometrics-related human age group classification. For texture categorization, we use CUReT and KTH-TIPS2a benchmark datasets. In the human age group classification, we consider the Images of Groups (IoG) database. The evaluation for both experiments is easily reproducible.

We perform evaluation using descriptors constructed by a random pick of pixel pairs, unsupervised learning using so called natural images [8], and finally, both unsupervised and supervised learning using images from the application. The motivation for using natural images comes from the earlier study where those together with Independent Component Analysis (ICA) were used to learn general purpose image descriptors. Like in [9], the aim is to evaluate whether powerful descriptors could also be produced by decision tree learning.

In our experiments, we only evaluate tree based structures leaving ferns for the future work.

4.1. Texture categorization

The CUReT dataset contains texture samples collected from 61 different materials from real-world surfaces with varying properties. We consider the publicly available cropped dataset [6] having a total of 5,612 images with 92 samples per each material class. For training image patches, we isolate a subset of 23 images per each material so that those images are put aside after learning. A descriptor is learnt by randomly sampling around 200,000 image patches of a certain size taking an even number of samples per each class. As this random process leads to a slightly different tree descriptor each time, we train 50 descriptors to gain statistically more relevant analysis. For the evaluation, we handle the remaining 69 images per class so that 23 per class is used for training a nearest-neighbor classifier and the remaining 46 samples are used for testing. We train the nearest-neighbor classifier 10 times randomly picking the 23 samples and then test it using the remaining 46 images. The mean accuracy of those ten iterations is then considered as the final performance of the descriptor. Throughout our texture categorization experiments we used L1 distance metric for the nearest-neighbor classifier.

In the CUReT examination, we learnt the descriptors using natural images, images picked from the application specific dataset, and application specific images together with the class information. The results, shown in Fig.1, validate that learning the descriptor with any level of supervision is more effective than randomly picking the pixel comparisons. They also show that in most of the cases, using application specific training patches is better than using natural images. Using the class information, however, does not seem to improve the accuracy but rather makes it worse.

Table 1. Mean accuracies of different descriptors on CUReT.

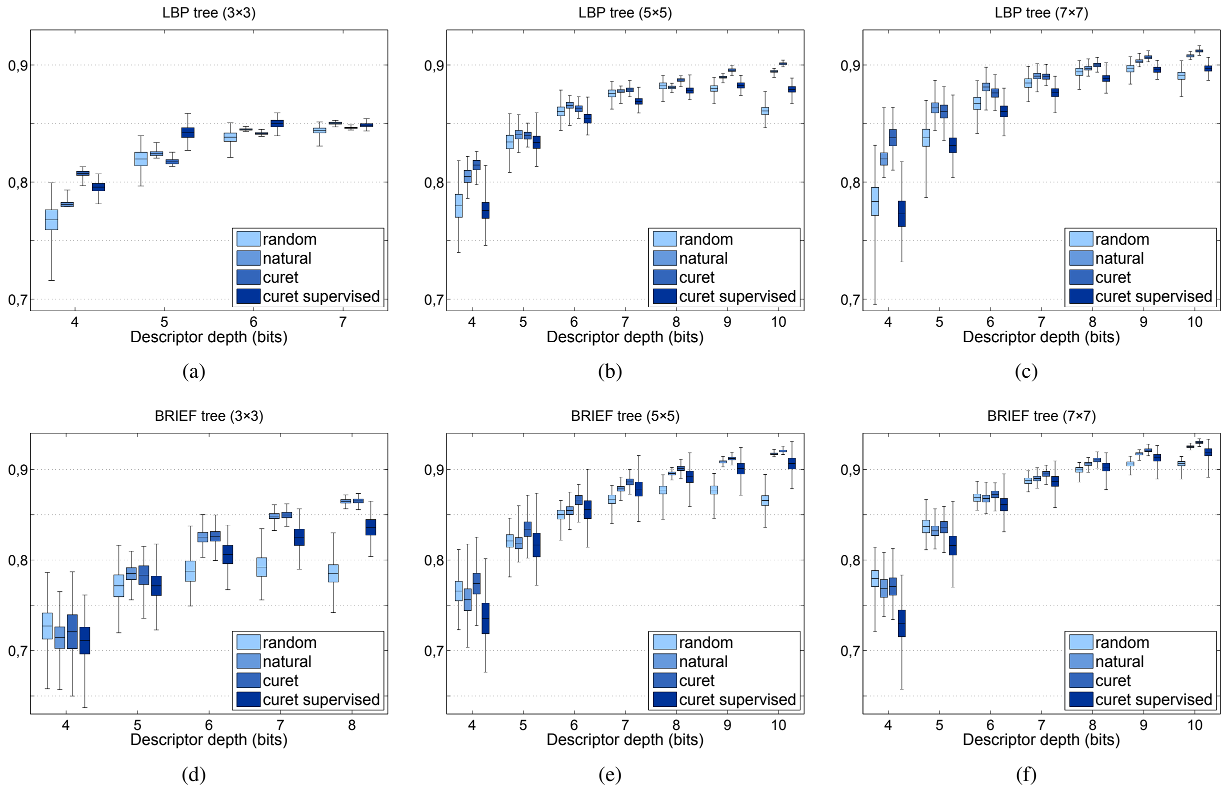
descriptor	mean accuracies
LBP $_{8,2}$ / $^{u2}_{8,2}$ / $_{10,2}$ / $^{u2}_{10,2}$	.889 / .864 / .849 / .874
LQP 32 / 64 / 128 / 256 / 512 / 1024	.856 / .865 / .892 / .897 / .902 / .901
$\text{LBP}^{\text{unsup}}_{\text{tree}}$ 32 / 64 / 128 / 256 / 512 / 1024	.840 / .863 / .879 / .887 / .896 / .901
$\text{BRIEF}^{\text{unsup}}_{\text{tree}}$ 32 / 64 / 128 / 256 / 512 / 1024	.834 / .866 / .886 / .901 / .912 / .920

We compared the best two methods on 5×5 neighborhoods against circular LBP and LQP using disk_5 pattern geometry with positive binary half [7]. The results, shown in Table 1, indicate that decision tree based descriptors are comparable to the benchmark methods. However, as can be seen in Fig.1, the results improve while using 7×7 neighborhood. Based on our experiments, especially LBP starts to substantially suffer from larger support area which is understandable as fixing 8 or 16 samples from a circular neighborhood, for example on a 7×7 neighborhood where there are $\binom{7\times7}{2}$ pixel pairs available, can not always provide the best set of those in the given application. Whereas for LQP, as also mentioned by the authors in [7], densely sampling larger support areas soon becomes infeasible ending up to the need for handcrafting the sampling strategy.

The KTH-TIPS2a dataset contains images of 11 materials, each of these are also present in the CUReT dataset [4]. Each material class contains images from four different material samples. For example, for the class *cracker*, samples of four different cracker qualities have been imaged. In addition, the dataset provides images with variations in scale, pose, and illumination. As a consequence, the KTH-TIPS2a dataset has a lot more within class variation in each material class than the CUReT dataset, yielding a more complicated setting. The dataset contains 4,608 samples which are divided so that 40 out of 44 samples have 108 images and the rest contain only 72 images.

For the KTH-TIPS2a experiment, we pick 8 to 9 images per a sample material so that finally there are 392 images. These images are used for learning the descriptors and then removed from further use like in the previous experiment. The rest of the experiment goes as in the original KTH-TIPS2a protocol: A nearest-neighbor classifier is trained by using three of the samples as a training set and then testing the classifier using the remaining sample. This procedure is repeated four times so that each sample is once used as a testing set. The final performance is then the mean of these four repetitions. Likewise in the CUReT experiment, we train a descriptor 50 times evaluating the performance for each round, and finally report means, standard deviations, and the whole range of accuracies.

The results, summarized in Fig.2, show that while using supervised setting in LBP-like sampling it is better to use larger neighborhoods. The same fact goes for BRIEF-like sampling. Most interestingly, the supervised setting with



(a)
(b)
(c)
(d)
(e)
(f)

Figure 1. Experimental results on the CUReT database using LBP and BRIEF-like sampling strategies with 3×3, 5×5 and 7×7 neighborhoods. (a), (b), and (c) illustrate the results using LBP-like sampling, and (d), (e), and (f) illustrate the results using BRIEF-like sampling. The colored boxes stand for standard deviations of 50 descriptor learning iterations, whereas the thin lines stand for the minimum and maximum accuracies.

BRIEF-like sampling seems to outperform other settings with a significant margin. In overall, the results clearly indicate the complexity of the KTH-TIPS2a dataset compared with the CUReT dataset.

Obviously, regarding both texture categorization experiments while using LBP-like sampling in supervised learning does not provide any improvement but rather makes the performance worse. However, there are some evidence that by enlarging the neighborhood could help which can be explained by the risen number of possible pixel comparisons. The reason why on CUReT the margin between supervised and unsupervised settings is far larger than on KTH-TIPS2a can be partly explained by a lot bigger number of different classes in CUReT. Perhaps it is just too complicated to learn a descriptor that manages to learn such a coding which succeeds in dividing the feature space as evenly as possible and, at the same time, separates each class as much as possible.

In Table 2, we compare the results of the best descriptors to LBP and LQP using different size of codebooks. We conducted the comparative analysis using the 5×5 neighborhood, as larger support areas seemed again to lower the performance for LBP, whereas for LQP, the implementation was infeasible due to the large size of the resulting look-up table.

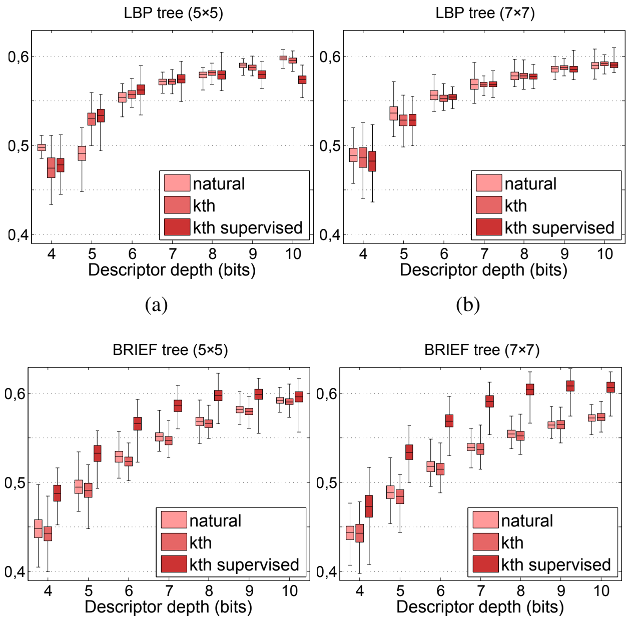
Table 2. Mean accuracies of different descriptors on KTH-TIPS2a.

descriptor	mean accuracies
LBP $_{8,2}$ / $^{u2}_{8,2}$ / $_{10,2}$ / $^{u2}_{10,2}$	.595 / .565 / .556 / .542
LQP 32 / 64 / 128 / 256 / 512 / 1024	.524 / .568 / .586 / .599 / .605 / .605
$\mathrm{LBP}^{\mathrm{unsup}}_{\mathrm{tree}}$ 32 / 64 / 128 / 256 / 512 / 1024	.530 / .558 / .572 / .582 / .588 / .596
$\mathrm{BRIEF}^{\mathrm{unsup}}_{\mathrm{tree}}$ 32 / 64 / 128 / 256 / 512 / 1024	.533 / .566 / .587 / .598 / .599 / .597

4.2. Age group classification

IoG consists of 28,231 facial images collected from Flickr images, taken in uncontrolled conditions. Each face is labeled with an age category defining seven age groups as follows: 0-2, 3-7, 8-12, 13-19, 20-36, 37-65, and 66+.

We first align each face with respect to eye coordinates provided by the database. We use a 64×64 pixels size of model to which all face images are fitted. Once normalized, the face is processed using the binary descriptor, and then, divided into 6×6 non-overlapping cells from



(a) (b)

(c) (d)

Figure 2. Experimental results on the `KTH-TIPS2a` database using different sampling strategies with 5×5 and 7×7 neighborhoods. (a) and (b) illustrate the results using LBP-like sampling, and (c) and (d) illustrate the results using BRIEF-like sampling.

which the descriptor labels are collected into a spatially enhanced histogram which is the final representation of the face. For classification, we train a multi-class Support Vector Machine (SVM) using a linear kernel and setting the cost $C = 1$.

The `IoG` database provides some predefined face sets that we utilize in this work: We use a predefined set of 3,500 face samples (500 faces per category) as a training the SVM. From the remaining faces we randomly collect 50 overlapping face sets which each contain 1,400 samples (200 faces per category). Then, in each descriptor learning round, we take 700 of those samples (100 faces per category) and learn the binary descriptors like in the previous experiments, sampling around 200,000 training image patches. The remaining 700 faces are then used in testing the performance of the age model.

The results of the age group classification experiment are shown in Fig. 3. Evidently, by growing the depth of the tree one is able to improve classification accuracy. The best result is got using unsupervised BRIEF-like descriptor learnt using application specific images. Nevertheless, learning by natural images provides almost equally good results. The most interesting is to notice the deteriorated performance of LQP descriptors. Also, LBP using *uniform patterns* makes the result worse. The large scale in accuracy for each descriptor reveals the fact that their performance is heavily dependent on the testing image sets. While the range in accuracies for tree-based descriptors seems to be very large, it is the same for LBP and LQP descriptors. To the best of our knowledge, the highest classification accuracies using the `IoG` database is reported in [2] (~ 0.56) and [21] (~ 0.52) which both use slightly different evaluation protocol yet report the results using only one fixed testing set. In that regard, we believe our results are more approriate than those in the reference studies.

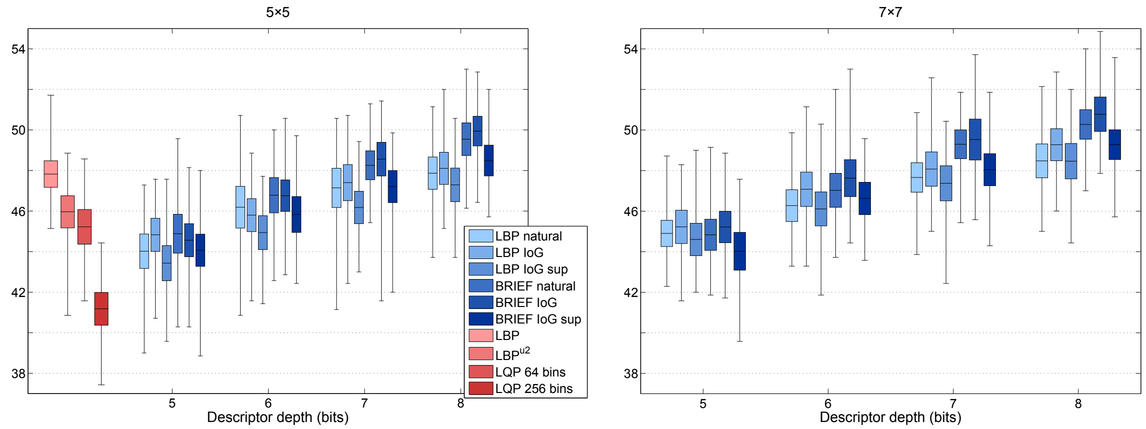
5. Discussion

In overall, the proposed framework has potential especially while using larger neighborhoods in which case we showed that better descriptors compared to LBP can be learned in the given applications. While the LQP descriptor remains a tempting method, using larger than 5×5 neighborhoods the reported release from handcrafting pixel comparisons becomes hard to meet because of the implementational issues. Moreover, we showed that using our framework one is able to produce competitive image descriptors without using application-specific data in the learning phase. These generic descriptors can be used as alternatives for conventional hand-crafted descriptors using equivalent or even shorter description lengths. The descriptors produced by our framework can be used in different applications in a similar manner as LBP or BRIEF and no large look-up tables are needed as in LQP.

6. Conclusions

In this paper we presented a unified framework for learning local image descriptors using binary decision trees. Our framework is inspired by the several previously presented binary descriptors such as LBP and BRIEF, and by the observation that those can be seen as a special cases of a more general decision tree model. We manifested three different levels of supervision under which the proposed framework was evaluated. These levels included unsupervised learning based on so called natural images, and unsupervised and supervised learning using application specific images. Regardless of the utilized supervision, the proposed framework constructs a tree-based descriptor which outputs a binary string for each pixel in a given image, which is then used to construct a histogram representation acting as the final description of the whole image.

We evaluated the performance of the descriptor learning framework by conducting experiments on two recognition problems, namely texture categorization and age group classification based on facial images. In texture categorization, we were able to learn descriptors that were at least comparable to the LBP and LQP reference methods, while in age group classification problem we demonstrated outperfoming results compared to the reference methods.



(a)

(b)

Figure 3. Experimental results on the `IoG` database using LBP and BRIEF-like descriptors with (a) 5×5 and (b) 7×7 neighborhoods.

Acknowledgement. The financial support of the Academy of Finland is gratefully acknowledged. This work has been partially performed within the EU FP7 project TABULA RASA (257289). The work was also partially supported by the FiDiPro program of Tekes.

References

[1] T. Ahonen and M. Pietikäinen. Image description using joint distribution of filter bank responses. *PRL*, 30(4):368–376, 2009. 2

[2] F. Alnajar, C. Shan, T. Gevers, and J.-M. Geusebroek. Learning-based encoding with soft assignment for age estimation under unconstrained imaging conditions. *IVC*, 30(12):946–953, 2012. 7

[3] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. BRIEF: Computing a local binary descriptor very fast. *IEEE TPAMI*, 34(7):1281–1298, 2012. 1, 2

[4] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Proc. ICCV*, pages 1597–1604, vol. 2, 2005. 5

[5] A. Criminisi and J. Shotton. *Decision forests for computer vision and medical image analysis*. Springer, 2013. 2

[6] K. J. Dana, B. van Ginneken, S. K. Nayar, and J. J. Koenderink. Reflectance and texture of real-world surfaces. *ACM TGRAPHICS*, pages 730–301, 1999. 5

[7] S. u. Hussain and B. Triggs. Visual recognition using local quantized patterns. In *Proc. ECCV*, pages 716–729, 2012. 3, 5

[8] A. Hyvärinen, J. Hurri, and P. O. Hoyer. *Natural Image Statistics*. Springer, 2009. 5

[9] J. Kannala and E. Rahtu. BSIF: Binarized statistical image features. In *Proc. ICPR*, pages 1363–1366, 2012. 2, 4, 5

[10] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE TPAMI*, 28(9):1465–1479, 2006. 2

[11] D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV'99*, pages 1150–1157 vol.2. 1

[12] T. Mäenpää, T. Ojala, M. Pietikäinen, and M. Soriano. Robust texture classification by subsets of local binary patterns. In *Proc. ICPR*, pages 947–950, 2000. 2

[13] D. Maturana, D. Mery, and A. Soto. Face recognition with decision tree-based local binary patterns. In *Proc. ACCV*, pages 618–629, 2010. 2, 3, 4

[14] F. Moosmann, E. Nowak, and F. Jurie. Randomized clustering forests for image classification. *IEEE TPAMI*, 30(9):1632–1646, 2008. 3, 4

[15] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002. 1

[16] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *Proc. ICISP*, pages 236–243, 2008. 1

[17] M. Özuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE TPAMI*, 32(3):448–461, 2010. 2, 3

[18] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen. *Computer Vision Using Local Binary Patterns*. Springer, 2011. 2

[19] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to SIFT or SURF. In *Proc. ICCV*, pages 2564–2571, 2011. 3

[20] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *Proc. CVPR*, pages 1–8, 2008. 2

[21] J. Ylioinas, A. Hadid, Y. Guo, and M. Pietikäinen. Efficient image appearance description using dense sampling based local binary patterns. In *Proc. ACCV*, pages 375–388, 2012. 7