

SIFT Descriptor for Binary Shape Discrimination, Classification and Matching

Insaf Setitra^{1,2}, Slimane Larabi²

¹ Research Center on Scientific and Technical Information Cerist, Algeria

² University of Science and Technology Houari Boumediene, Algeria
isetitra@cerist.dz,slarabi@usthb.dz

Keywords: SIFT, Shape description, Classification, Image retrieval.

Abstract. In this work, we study efficiency of SIFT descriptor in discrimination of binary shapes. We also analyze how the use of 2 – *tuples* of SIFT keypoints can affect discrimination of shapes. The study is divided into two parts, the first part serves as a primary analysis where we propose to compute overlap of classes using SIFT and a majority vote of keypoints. In the second part, we analyze both classification and matching of binary shapes using SIFT and Bag of Features. Our empirical study shows that SIFT although being considered as a texture feature, can be used to distinguish shapes in binary images and can be applied to the classification of foreground’s silhouettes.

1 Introduction

Scale Invariant Feature Transform (SIFT) proposed by David Lowe (2004) [1] is an approach for detecting and extracting local feature descriptors invariant to rotation and scaling and becomes a useful feature for image representation. Invariance of SIFT is insured by filtering and subsampling an original image so that only strong keypoints of the image are kept. To do so, difference of Gaussian is computed by differentiating each two consecutive filtered images by a Gaussian filter at different scales and repeating the process by subsampling the original image on different octaves. The final descriptor is then a histogram of gradient magnitude and orientation of each keypoint detected. Advantages of SIFT are then primarily its invariance to scale and rotation. Moreover, its strength resides in its locality. Since its introduction, SIFT received a high popularity and has been largely used in many applications [3], [4], [5], [6],[7]. However, since the descriptor is local and based on gradient magnitude and orientation of keypoints, it was considered a texture feature [2] and has received very few interest when applied to binary images. Instead, state of the art on binary image classification and matching use more shape features than texture features [18], [8], [9], [10], [11]. The most relevant work dealing with SIFT on binary images is devoted to hand gestures classification [12] by the use of SIFT feature applied to binary masks. Authors claim to improve accuracy of classification and to decrease time processing since the descriptor is not sensitive to illumination changes and

less information is needed to quantify the keypoints. In this work, we study the discrimination of SIFT feature on binary images which has no texture. The classification of shapes and their matching is also studied using one keypoint or 2 – *tuples* keypoints. While the descriptor is local and represented by a set of keypoints, and number keypoints is different from an image to another, supervised classification algorithms might fall. In order to overcome this issue, we follow same rationale as in [21] i.e. we use Bag of Features approach to make a new representation of the features which simplify the task of classification.

This paper is organized as follows: we present in section 2 an approach based on a majority vote of keypoints to compute overlap of SIFT over different classes to study discrimination of SIFT on binary images. Section 3 is devoted to classification of binary images using SIFT and Bag of Features. We present our results in section 4. We conclude this paper by some future works.

2 Class Overlap analysis: Approach and Results

2.1 The approach

In order to analyze discrimination of SIFT feature applied to binary images for classification, we first perform an empirical study of overlap between classes using SIFT. In a first study, we consider each keypoint independently of its neighbors. In a second study, we add the notion of neighboring where, instead of considering each keypoint independently, we consider 2 neighboring tuples of SIFT keypoints and compute the overlap of classes. SIFT descriptor is computed in the same way as in [1] and results in a set of keypoints each of which is a 128 dimensional vector of gradient orientations and a 3 dimensional vector of row coordinate, column coordinate and radius of the keypoint. Neighboring 2 – *tuples* of SIFT is a set of 256 dimensional vector (128×2), where the first 128 values are gradient orientation of the first keypoint and the second 128 values are gradient orientation of the neighboring keypoint. The neighbor of a keypoint is chosen so that its row and column coordinates are the closest to its neighbor.

To compute overlap between classes, we choose m objects from each class and compute their SIFT features. To each SIFT (respectively 2-tuples SIFT) keypoint is assigned a label which represents the class of the image to which it belongs to. Overlap between all classes is initialized to 0. Each keypoint of each image is then compared to all keypoints of all images, and the keypoint which gave the minimum distance is chosen. Overlap between the keypoint class to be compared and the class of keypoint which gave the minimum distance is incremented. Finally, overlap percentage is computed. The same process is applied for 2 – *tuples* of keypoints. The following algorithm gives an insight of the process.

Algorithm 1 sorting keypoints

```
1: BEGIN
2:  $Overlap(C_i, C_j) = 0, i, j = 1..numberClasses$ 
3: for each image  $I$  of the dataset do
4:    $C_i$  is the class of  $I$ ;
5:   for Each Keypoint  $K$  of image  $I$  do
6:     for all keypoints  $K'$  of all images of the dataset different than  $I$  including also
       images of class  $C_i$  do
7:       Compute Euclidean Distance between  $K$  and  $K'$ 
8:     end for
9:     Get  $K'^*$  the keypoint form all  $K'$  which gave the minimum distance
10:    Get  $C_j$  class of image to which belong  $K'^*$ 
11:     $Overlap(C_i, C_j) ++$ ;
12:  end for
13: end for
14:  $Overlap(C_i, C_j) = \frac{Overlap(C_i, C_j)}{\sum_{i=1}^{numberClasses} Overlap(C_i, i)}$ 
15: END
```

2.2 Overlap accuracy of the dataset ETH-80

Overlap experiments are performed over 100 images from each of the 8 classes of ETH-80 which is a dataset composed of 400 images of 8 classes: apple, car, cow, cup, dog, horse, pear and tomato. Each of the 400 images of each class are obtained using 10 distinct objects of the same class taken using a spherical rotation of the camera around the object while keeping the same distance from the camera and the object being pictured. Table 1 shows that SIFT keypoints always reflect the true class label. Indeed, the highest percentage of keypoints on classes is always relative to the true class label. However, some keypoints still vote for a different class. The highest the percentage is for another class different than the real one, the most similar are the classes. For example, in the first line, the highest overlap of keypoints for the class 1 is 58.52% with the class 1 which represents the class apple. The second highest overlap percentage is given to class 8 which is the class tomato. This means that, 22.92% of keypoints present in the class apple are also present in the class tomato which is true when we look at their two shapes. The same happens to the class 3 (cow) with the class 5 and 6 (dog and horse) in the third line of the table. With high percentage of overlap for the class 3 with the class 5 (13%) and the class 6 (18%) while few percentages are observed on other classes. The table, although reflects some overlaps between classes, can prove discrimination of SIFT feature on binary images.

In the second experiment, comparison between overlap of classes using independent SIFT keypoints and 2-tuples of SIFT keypoints is performed. Experiment is performed over 80 images where 10 images of each class are chosen to compute the overlap. Table 1 shows that, although 2-tuples of SIFT is still discriminating as percentages in the diagonal are still the highest, discrimination is decreased compared to overlap when using SIFT keypoints independently. Only

three classes where discrimination is improved are class 3 (cow), class 4 (cup) and class 5 (dog) with improvement of 10.62%, 13.75% and 0.90% respectively. These percentages are presented in table 2 where positive values represent increase in discrimination and negative values represent decrease in discrimination. Overall improvement using $n-tuples$ of SIFT is 1.08% which is negligible knowing that time processing is highly increased. At this point, we put the hypothesis that $2-tuples$ SIFT do not improve discrimination between shapes and perform more experiments below to verify this hypothesis.

Table 1. Overlap computed using SIFT, $2-tuples$ of SIFT and $2-tuples$ of SIFT descriptor respectively over 100, 10 and 10 images respectively of each of the classes: C_1 = 'apple', C_2 = 'car', C_3 = 'cow', C_4 = 'cup', C_5 = 'dog', C_6 = 'horse', C_7 = 'pear', and C_8 = 'tomato'.

| | | | | | | | | |
|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | $O_1\%$ | $O_2\%$ | $O_3\%$ | $O_4\%$ | $O_5\%$ | $O_6\%$ | $O_7\%$ | $O_8\%$ |
| C_1 | 58.52 | 3.63 | 1.04 | 4.93 | 1.06 | 0.62 | 7.27 | 22.92 |
| C_2 | 10.53 | 47.86 | 6.23 | 8.16 | 4.93 | 5.72 | 6.80 | 9.77 |
| C_3 | 4.46 | 8.40 | 43.76 | 6.12 | 13.48 | 18.05 | 2.90 | 2.84 |
| C_4 | 5.43 | 2.39 | 1.33 | 76.84 | 2.01 | 1.36 | 5.12 | 5.53 |
| C_5 | 4.70 | 6.82 | 13.38 | 7.89 | 42.13 | 16.45 | 4.99 | 3.65 |
| C_6 | 4.41 | 6.21 | 17.58 | 5.18 | 15.57 | 43.21 | 4.88 | 2.96 |
| C_7 | 14.95 | 2.92 | 0.85 | 7.32 | 1.14 | 1.64 | 58.55 | 12.63 |
| C_8 | 25.68 | 3.85 | 1.29 | 5.65 | 1.17 | 1.26 | 6.75 | 54.35 |
| | $O_1\%$ | $O_2\%$ | $O_3\%$ | $O_4\%$ | $O_5\%$ | $O_6\%$ | $O_7\%$ | $O_8\%$ |
| C_1 | 51.10 | 3.40 | 0.47 | 4.55 | 0.60 | 0.66 | 7.38 | 31.84 |
| C_2 | 14.05 | 30.18 | 3.96 | 12.12 | 5.15 | 3.61 | 12.41 | 18.51 |
| C_3 | 5.52 | 7.85 | 38.25 | 8.41 | 11.37 | 21.37 | 4.55 | 2.69 |
| C_4 | 4.91 | 3.53 | 1.67 | 77.80 | 2.03 | 1.58 | 3.23 | 5.25 |
| C_5 | 6.72 | 8.03 | 14.51 | 10.03 | 30.11 | 18.01 | 7.98 | 4.60 |
| C_6 | 8.00 | 5.44 | 23.21 | 7.57 | 15.86 | 30.33 | 5.68 | 3.91 |
| C_7 | 17.89 | 6.18 | 1.04 | 3.45 | 1.16 | 1.63 | 51.35 | 17.30 |
| C_8 | 33.62 | 4.07 | 0.39 | 5.32 | 0.97 | 0.72 | 8.27 | 46.63 |
| | $O_1\%$ | $O_2\%$ | $O_3\%$ | $O_4\%$ | $O_5\%$ | $O_6\%$ | $O_7\%$ | $O_8\%$ |
| C_1 | 60.59 | 5.40 | 0.90 | 5.40 | 1.26 | 0.42 | 5.76 | 20.25 |
| C_2 | 18.61 | 31.40 | 4.31 | 11.64 | 7.29 | 3.50 | 8.46 | 14.80 |
| C_3 | 7.13 | 8.55 | 27.63 | 11.59 | 16.85 | 20.02 | 3.85 | 4.38 |
| C_4 | 8.95 | 4.33 | 2.92 | 64.06 | 2.96 | 1.53 | 4.94 | 10.32 |
| C_5 | 5.87 | 10.81 | 18.46 | 9.86 | 29.21 | 15.88 | 5.93 | 3.99 |
| C_6 | 6.75 | 5.69 | 18.75 | 7.05 | 14.96 | 35.59 | 8.94 | 2.27 |
| C_7 | 16.37 | 4.50 | 1.60 | 6.81 | 1.74 | 3.49 | 52.96 | 12.54 |
| C_8 | 25.90 | 5.44 | 1.62 | 9.67 | 0.83 | 0.82 | 6.16 | 49.55 |

Table 2. Overlap Difference between 2-tuples SIFT overlap (table 1 part 2) and SIFT overlap (table 1 part 3) for classes: C_1 ='apple', C_2 ='car', C_3 ='cow', C_4 ='cup', C_5 ='dog', C_6 ='horse', C_7 ='pear', and C_8 ='tomato'.

| | $O_1\%$ | $O_2\%$ | $O_3\%$ | $O_4\%$ | $O_5\%$ | $O_6\%$ | $O_7\%$ | $O_8\%$ |
|-------|-------------|-------------|--------------|--------------|-------------|-------------|-------------|-------------|
| C_1 | -9.49 | 2.01 | 0.43 | 0.85 | 0.66 | -0.23 | -1.62 | -11.59 |
| C_2 | 4.56 | -1.21 | 0.34 | -0.47 | 2.14 | -0.11 | -3.95 | -3.70 |
| C_3 | 1.61 | -0.70 | 10.62 | 3.18 | 5.48 | -1.35 | -0.70 | 1.70 |
| C_4 | 4.04 | -0.80 | 1.25 | 13.75 | 0.93 | -0.06 | 1.71 | 5.08 |
| C_5 | -0.85 | -2.78 | 3.95 | -0.17 | 0.90 | -2.13 | -2.06 | -0.62 |
| C_6 | -1.25 | -0.25 | -4.46 | -0.52 | -0.90 | -5.26 | 3.26 | -1.64 |
| C_7 | -1.53 | 1.68 | 0.55 | 3.36 | 0.58 | 1.86 | -1.61 | -4.75 |
| C_8 | -7.72 | -1.37 | 1.22 | 4.35 | -0.14 | 0.09 | -2.10 | -2.92 |

3 Classification of binary images using SIFT and Bag of Features.

Classification of images using SIFT is very difficult because of the high dimensionality of SIFT feature and the non-fixed feature representation. We use the Bag of Features method [19] to overcome this issue and present it in what follows.

3.1 Image representation using Bag of Features

After computed, SIFT features for all images of the training set are concatenated into one matrix $M(n \times 128)$ where $n = \sum_{i=1}^m n_i$ and n_i is number of keypoints of image I_i and m is the number of images of the training set. Rows of the matrix M are then clustered using K-means clustering algorithm. K-means is an iterative algorithm which starts by defining l centers of clusters randomly from the set of n features. Then, the centers are updated by calculating distance between the center and remaining features. Once features are assigned to a center, the mean of the cluster is chosen as the new center. Update is repeated until one of two conditions: either maximum number of iterations is reached or centers are stable (means of clusters converge). Result of clustering is then l stabilized clusters, each cluster is represented by its center which is one of the 128 dimensional vector of keypoints. Let c_i be center of the cluster i . Visual vocabulary is then the matrix $VC = c_1, c_2 \dots c_l$. The following step consists of representing each image of the dataset (training and test) with its Bag of Features using visual vocabulary. To do so, for each keypoint of each image, we compute the Euclidean distance with each Center of the Visual Vocabulary. The minimum Euclidean distance reflects the center chosen for the keypoint. Number of occurrences of a center is computed and represented by a normalized histogram of dimension k (number of features in the visual Vocabulary). Each image is then represented by its histogram of visual Features also referred as Bag of Features.

3.2 Training and classification using Bag of Features

After describing training and test images by their Bag of Features, the following step consists of attributing a class label to each image of the test set. The class label is given to the nearest neighbor of the test image and the training image. In other words, for an image I of the test set, the distance between its Bag of Features and Bag of Features of all images in the training set is computed. The Bag of Features which gave the minimum distance is chosen and its class label is given to the image I . Nearest Neighbor is one of the simplest supervised classification algorithms but is still accurate. We use χ^2 distance as it showed to give better results. χ^2 distance between two histograms is described as follows:

$$\chi^2 = \sum_{i=1}^l \frac{(HI_i - HT_i)^2}{HT_i} \quad (1)$$

where l is the dimension of the feature vector (number of bins), HI_i the Bag of Features of image I_i and HT_i Bag of Features of an image T in the training set.

4 Experimental Results

In this section we evaluate effectiveness of bag of SIFT features applied to binary images for object classification and matching. We present the quantitative analysis of the method applied to binary images on both ETH-80 [15] and MPEG-7 Core Experiment CE-Shape-1 data set [16] datasets. MPEG-7 dataset consists of 1,400 images grouped into 70 classes. Each class has 20 different shapes.

Difference between ETH-80 and MPEG-7 datasets is that the latter contains only binary images while the former contains both binary and RGB images of the same objects. We use this property in order to enhance our experiments and prove effectiveness of SIFT feature on binary images and compare accuracy of our classifier on binary and same RGB images. Besides, we divided the MPEG-7 data set into two sets: training set and test set. For each class, 10 shapes are chosen as the training samples and the remaining 10 shapes are then used for testing. We do the same for ETH-80, however, since this dataset contains much more images of the same class than MPEG-7, we vary size of the training and analyze accuracy at each time. Finally, we analyze matching using SIFT. To do so, we extend our Nearest Neighbor Classification algorithm to make it return instead of a class label, the m nearest neighboring images of a query image. We compare then resulting matching with some state of the art techniques: Inner Distance (IDSC) [13] and Graph Transduction [18], shape Context (SC) and partial matching [11]). For the qualitative analysis of matching, we input to our system the same query images as the ones of methods we compared with and display our results with their ones. We present in following subsection experiments done on both datasets. For the quantitative analysis of matching, we compute the bull's eye score of matching.

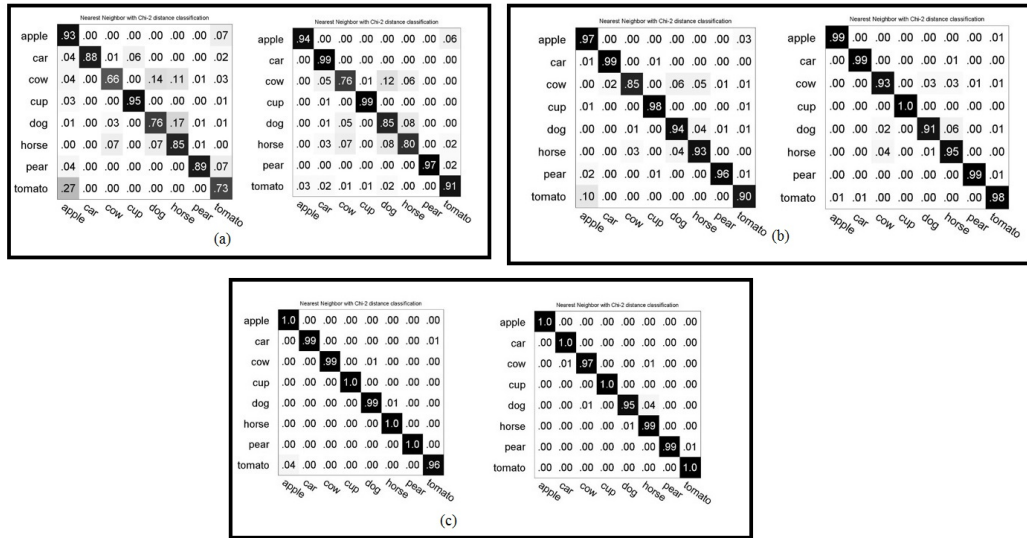


Fig. 1. Confusion matrices for ETH-80 with : (a) (20% and 80%), (b) (40% and 60%), (c) (60% and 40%) (training and test set size) respectively (left matrices binary images, right matrices RGB images).

4.1 Classification accuracy of the dataset ETH-80

For classification accuracy analysis, extensive experiments of ETH-80 are done by dividing the whole dataset into training and classification. Division is done following several percentages from 10 to 70% where this percentage represents ratio of training set with the whole dataset. In a first experiment, we vary size of both training and test set where sum of their two ratios is equal to 100% (upper part of table 3). By doing this, we insure not to include images of the training set into the test set. In a second experiment, we choose a small ratio of the test set (30%) and fix it for all percentages of training. Size of the training set in both cases do not exceed 70% which represent the classical percentage used in most classification experiments (2/3 training and 1/3 test). We analyze classification accuracy of RGB images and their binary masks (binary images) using the same parameters of the code book (same number of iterations of K-means clustering, same number of features in the visual vocabulary, same distance measure). Table 3 shows that classification accuracy of RGB images in most cases outperforms the one of binary images of the original RGB images. However, the difference is slight and in one case, classification of binary images outperformed classification of RGB images. In order to analyze more deeply the classification results, we moreover generate for each ratio confusion matrices. We generate more than 60 confusion matrices by varying the size of both training and test set, however, we kept only the most significant ones which correspond to ratios of table 3. Confusion matrices show that classification accuracy is enhanced in RGB images

compared to binary images, however, the difference is still slight. Some interesting points can be seen in the confusion matrices: Confusion of RGB images can occur even when the shape is different, example of such a confusion can be seen in figure 1 where horses have been confused with cows. This confusion is less apparent in the binary images and accuracy of horse classification can reach 100% for binary images while it does not for RGB images. Such errors occurred for horses with same colors as cows and dogs and when some animals have same skin (texture). The same observation can be made for classes "Dog" and "Cow". For all other classes, classification accuracy of RGB images is slightly higher than classification accuracy of the same binary images.

The last experiment performed on ETH-80 is done in order to analyze the contribution of 2 – tuples SIFT. Experiment emphasizes hypothesis posed in 2.1 where figure 2 shows that considering 2-tuples of SIFT decreases the classification accuracy. Overall accuracy using SIFT keypoints independently is 90.19% while it decreases when using neighboring 2-tuples of SIFT to 82.82%. Indeed, when considering neighboring keypoints, no a priori information about the contour is considered. Also, processing time when processing 2 – tuples is increased compared to independent SIFT keypoints. We do not perform more experiments on 2 – tuples SIFT as primary experiments show decrease in accuracy.

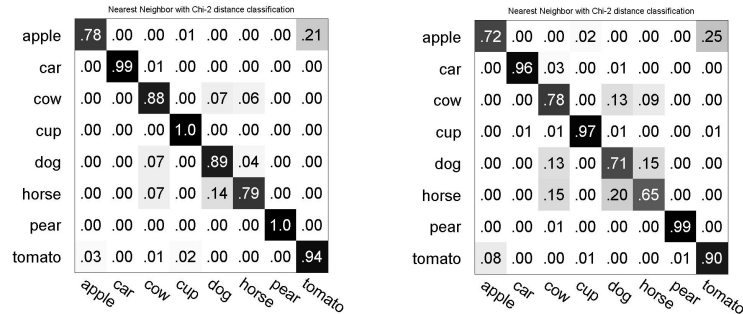


Fig. 2. Confusion matrices for ETH-80 with size 50% training and 50% test. Left matrix classification without 2 – tuples SIFT, right matrix classification with 2 – tuples SIFT).

Table 3. Overall accuracy on ETH-80 dataset by varying size of training set and test set

| Training % | Test % | Accuracy % | |
|------------|--------|------------|-------|
| | | Binary | RGB |
| 20 | 80 | 83.01 | 90.22 |
| 40 | 60 | 93.91 | 96.79 |
| 60 | 40 | 99.03 | 98.71 |
| | | | |
| 10 | 30 | 77.77 | 84.61 |
| 20 | 30 | 84.82 | 88.35 |
| 30 | 30 | 90.49 | 93.05 |
| 40 | 30 | 92.20 | 94.87 |
| 50 | 30 | 94.23 | 95.51 |
| 60 | 30 | 96.68 | 97.75 |
| 70 | 30 | 99.14 | 99.25 |

4.2 Classification and matching accuracy of the dataset MPEG-7

For MPEG-7 dataset, 10 images from each of the 70 classes of MPEG-7 are used for training and 10 images are used for tests. As Nearest Neighbor does not need any cross validation, we directly compute classification after training. Confusion matrix is presented in figure 3. The confusion matrix is less comprehensible because number of classes is very high, however, we can see from blue rectangles in the diagonal classes that are confused the most. These classes are: 11 (camel), 28 (device.8), 69(turtle), 70 (watch). Overall classification was 78%. Figure 4 (left) shows that worst results of SIFT were obtained with the query "bird" and "lizzard". For the class "lizzard", confusion due to the presence of keypoints of these two classes in many other classes. Confusion of "bird" can be explained by the presence of very few keypoints in shapes of that class while shapes of other classes have much more keypoints including the ones present in the class "bird". Right part of figure 4 in another hand is way much encouraging, where SIFT present competing results compared to state of the art methods.

Table 4 presents retrieval rate for the MPEG-7 database. Retrieval rate is measured by the so-called bullseye score which counts all matching objects within the 40 most similar candidates [20]. While some classes give perfect retrieval results (100%), some other classes reach a minimum rate of 15%.

5 Conclusion

We have presented in this work an empirical study of usefulness of SIFT feature applied to binary images. Experiments show that although SIFT is most often

Table 4. Bull's score for MPEG-7 dataset

| | | | |
|---------------|----------------|----------------|----------------|
| Bone: 95% | bird: 15% | children: 100% | device4:40% |
| flatfish: 90% | jar: 60% | snake: 20% | Comma: 90% |
| bottle: 55% | chopper: 100% | device5: 45% | fly: 65% |
| key: 30% | shoe: 100% | Glas: 90% | brick: 75% |
| classic: 50% | device6: 55% | fork: 20% | lizzard: 60% |
| spoon: 20% | HCircle: 100% | butterfly: 40% | crown: 95% |
| device7: 40% | fountain: 100% | lmfish: 30% | spring: 65% |
| Heart: 100% | camel: 60% | cup: 90% | device8: 40% |
| frog: 30% | octopus: 25% | stef: 55% | Misk: 100% |
| car: 95% | deer: 50% | device9: 45% | guitar: 30% |
| pencil: 20% | teddy: 100% | apple: 95% | carriage: 100% |
| device0: 50% | dog: 35% | hammer: 30% | car: 40% |
| tree: 80% | bat :45% | cattle: 50% | device1: 30% |
| elephant: 45% | hat: 75% | pocket: 90% | truck: 100% |
| beetle: 35% | cellular: 100% | device2: 30% | face:100% |
| horse: 35% | rat: 95% | turtle: 15% | bell: 95% |
| chicken: 20% | device3: 35% | fish: 35% | horseshoe: 85% |
| ray: 45% | watch: 80% | | |

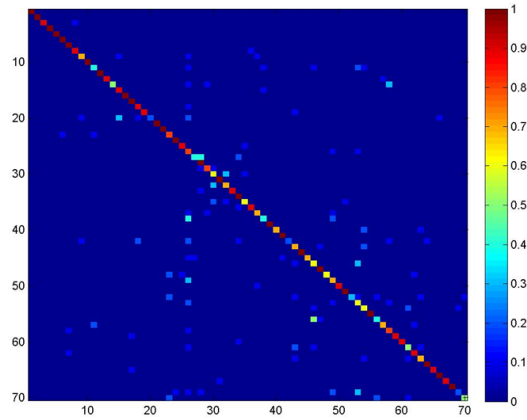


Fig. 3. Confusion matrix plot for MPEG-7 dataset with training size 1/2, test size 1/2.

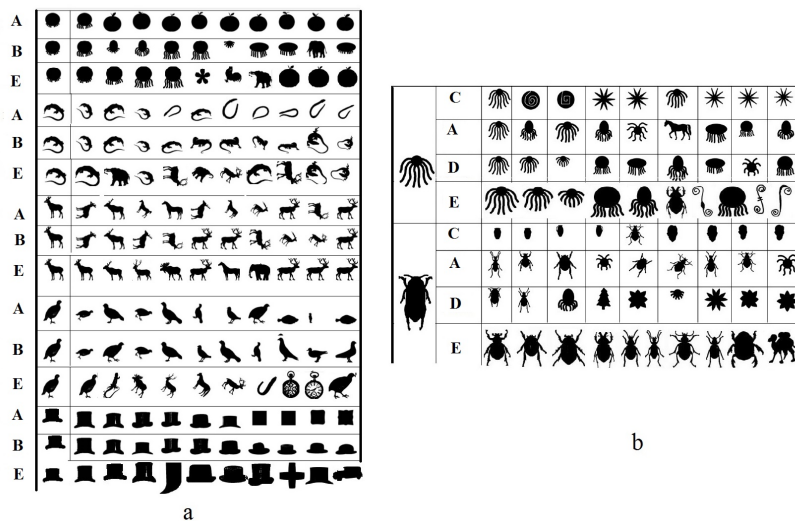


Fig. 4. (a) Matching comparison of our proposed study (E in the figure) with Inner Distance (IDSC) (A in the figure) [13] and Graph Transduction (B in the figure) [18], (b) Matching comparison of our proposed study (E in the figure) with Inner Distance (IDSC) (A in the figure) [13], Shape Context (SC) (C in the figure) and partial matching (D in the figure) [11]

dedicated to be applied on texture images, it can still be applied on binary images and can sometimes outperform results obtained using RGB and grey level images. We aim in further works to apply SIFT to classify directly masks derived from background subtraction [14].

References

1. Lowe, David G., Distinctive Image Features from Scale-Invariant Keypoints, *Int. J. Comput. Vision*, 60(2), pp. 91–110, 2004.
2. Wang, Xiaogang Intelligent Multi-camera Video Surveillance: A Review, *Pattern Recogn. Lett.*, 34(1), pp. 3–19, 2013.
3. Tsuchiya, M. and Fujiyoshi, H., Evaluating Feature Importance for Object Classification in Visual Surveillance, 18th International Conference on Pattern Recognition, 2006. ICPR 2006.
4. Zheng Song and Qiang Chen and Zhongyang Huang and Yang Hua and Shuicheng Yan, Contextualizing object detection and classification, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
5. Zhaoxiang Zhang and Min Li and Kaiqi Huang and Tieniu Tan, Boosting local feature descriptors for automatic objects classification in traffic scene surveillance, 19th International Conference on Pattern Recognition, 2008. ICPR 2008.
6. Deselaers, Thomas and Heigold, Georg and Ney, Hermann, Object Classification by Fusing SVMs and Gaussian Mixtures, *Pattern Recogn.*, 43(7), pp.2476–2484, 2010.

7. Cristina Conde and Daniela Moctezuma and Isaac Martn De Diego and Enrique Cabello, HoGG: Gabor and HoG-based human detection for surveillance in non-controlled environments, *Neurocomputing*, 100(0), pp. 19 - 30, 2013.
8. Chahooki, Mohammad Ali Zare and Charkari, Nasrollah Moghaddam, Shape Classification by Manifold Learning in Multiple Observation Spaces, *Inf. Sci.*, 262, pp.46–61, March, 2014.
9. Loris Nanni and Alessandra Lumini and Sheryl Brahmam, Ensemble of different local descriptors, codebook generation methods and subwindow configurations for building a reliable computer vision system, *Inf. Sci.*, 26(2), pp.89 - 100, 2014.
10. Lorenzo Torresani and Martin Szummer and Andrew Fitzgibbon, Efficient Object Category Recognition using Classemes, *European Conference on Computer Vision (ECCV)*, pp. 776–789, 2010.
11. Bouagar, Saliha and Larabi, Slimane, Efficient descriptor for full and partial shape matching, *Multimedia Tools and Applications*, pp. 1-23, doi=10.1007/s11042-014-2417-0, 2014.
12. Lin, Wei-Syun and Wu, Yi-Leh and Hung, Wei-Chih and Tang, Cheng-Yuan, A Study of Real-Time Hand Gesture Recognition Using SIFT on Binary Images, *Advances in Intelligent Systems and Applications - Volume 2*, pages=235-246.
13. Haibin Ling and Jacobs, D.W., Shape Classification Using the Inner-Distance, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol 29(2), pp. 286-299.
14. Setitra, I. and Larabi, S., Background Subtraction Algorithms with Post-processing: A Review, *22nd International Conference on Pattern Recognition (ICPR)*, 2014.
15. Leibe, B. and Schiele, B., Analyzing appearance and contour based methods for object categorization, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.
16. Longin Jan Latecki and Rolf Lakmper and Ulrich Eckhardt, Shape Descriptors for Non-rigid Shapes with a Single Closed Contour, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000.
17. Xiang Bai and Cong Rao and Xinggang Wang, Shape Vocabulary: A Robust and Efficient Shape Representation for Shape Matching, *IEEE Transactions on Image Processing*, 23(9), pp. 3935-3949, 2014.
18. Xiang Bai and Yang, Xingwei and Latecki, L.J. and Wenyu Liu and Zhuowen Tu, Learning Context-Sensitive Shape Similarity by Graph Transduction, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(5), pp. 861-874, 2010.
19. Seidenari, L. and Serra, G. and Bagdanov, A.D. and Del Bimbo, A., Local Pyramidal Descriptors for Image Recognition *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014, 36(5), pp. 1033-1040, 2013.
20. Kotschieder, Peter and Donoser, Michael and Bischof, Horst, Beyond Pairwise Shape Similarity Analysis, *Computer Vision ACCV 2009*, pp. 655-666, 2009.
21. Bharath Ramesh Peter and Chang Xiand and Tong Heng Lee Shape classification using invariant features and contextual information in the bag-of-words model *Pattern Recognition*, vol. 48, n. 3, pp. 894-906, 2015.